UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JORGE MAGNO LOPES MORAES

DDSHP: UM SISTEMA PARA A DETECÇÃO DE DDOS EM IOT BASEADO NO PARÂMETRO DE HURST E SDN

FORTALEZA

2023

JORGE MAGNO LOPES MORAES

DDSHP: UM SISTEMA PARA A DETECÇÃO DE DDOS EM IOT BASEADO NO PARÂMETRO DE HURST E SDN

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Redes de Computadores.

Orientador: Prof. Dr. Arthur de Castro Callado.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M821d Moraes, Jorge Magno Lopes.
DDSHP: Um Sistema para a Detecção de DDoS em IoT baseado no Parâmetro de Hurst e SDN / Jorge Magno Lopes Moraes. – 2023.
79 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2023.
Orientação: Prof. Dr. Arthur de Castro Callado.

1. Internet das coisas. 2. Segurança. 3. Rede definida por software. 4. Parâmetro de hurst. 5. Negação de serviço distribuída. I. Título.

CDD 005

JORGE MAGNO LOPES MORAES

DDSHP: UM SISTEMA PARA A DETECÇÃO DE DDOS EM IOT BASEADO NO PARÂMETRO DE HURST E SDN

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Redes de Computadores.

Aprovada em: 24/11/2023.

BANCA EXAMINADORA

Prof. Dr. Arthur de Castro Callado (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Miguel Franklin de Castro
Universidade Federal do Ceará(UFC)

Prof. Dr. Carlos Alberto Kamienski
Universidade Federal do ABC (UFABC)

Aos meus pais, Raimundo Jorge Lima Moraes e Maria Luzilene Bezerra Lopes. O caminho foi desafiador, mas com o amor e apoio de vocês, tudo deu certo. Vocês são a minha fonte inesgotável de apoio, amor e inspiração ao longo de toda a minha jornada acadêmica e de vida.

AGRADECIMENTOS

À Instituição Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro com a concessão da bolsa de estudo de Mestrado, n.º de processo 131102/2020-6.

Ao Prof. Dr. Arthur de Castro Callado pela orientação excepcional ao longo deste processo. Sua disponibilidade para esclarecer dúvidas e fornecer orientação técnica foram aspectos fundamentais que enriqueceram minha jornada acadêmica. Agradeço pelo comprometimento demonstrado, pela paciência incansável e pela inspiração constante, que foram fundamentais para o alcance dos resultados obtidos.

Aos ilustres professores que compuseram a banca examinadora, o Prof. Dr. Miguel Franklin de Castro e o Prof. Dr. Carlos Alberto Kamienski. Agradeço sinceramente pelo tempo dedicado à avaliação do meu trabalho, bem como pelas inestimáveis colaborações e sugestões oferecidas ao longo do processo.

Aos meus colegas de mestrado, Francisco Luciano Castro Martins Júnior e Victória Tomé de Oliveira. Ao Luciano, meu profundo agradecimento por sua generosidade em compartilhar conhecimento e por estar sempre disponível para esclarecer minhas dúvidas ao longo deste desafiador caminho. Quanto à Victória, além de ser uma amiga querida, ela se revelou uma confidente valiosa, compartilhando não apenas as alegrias, mas também enfrentando as mesmas dificuldades que surgiram durante o processo. Agradeço a ambos pela companhia, pelo apoio mútuo e pela contribuição significativa para tornar essa jornada mais enriquecedora e gratificante e menos tortuosa.

Aos amigos incríveis que estiveram ao meu lado durante esta jornada acadêmica desafiadora. Agradeço sinceramente ao Lucas Lima Mota, Areta Lourenço Rodrigues e Francisca Caroline Albuquerque Costa por serem fontes constantes de apoio e alegria. Mesmo sem saberem, a presença de vocês foi uma fonte constante de motivação para seguir em frente. Em nossas reuniões e conversas, encontrava não apenas momentos de descontração, mas também um verdadeiro refúgio. Além disso, dedico um agradecimento especial à minha namorada, Crislayne Vieira de Araújo, por ser uma fonte constante de apoio inabalável e por trazer luz e alegria aos meus dias. Seu incentivo silencioso foi como um farol, iluminando meu caminho e dando-me a força necessária para persistir. A todos vocês, meu sincero agradecimento por fazerem parte desta jornada e por tornarem cada desafio mais leve e cada conquista mais significativa.

Aos meus queridos mãe, Maria Luzilene Bezerra Lopes, e pai, Raimundo Jorge Lima

Moraes, palavras parecem pequenas diante da imensidão do meu amor e gratidão por vocês. Ao longo da minha jornada, cada passo foi sustentado pelos alicerces que vocês construíram para mim. Desde os primeiros ensinamentos sobre a vida até o calor do amor incondicional que sempre emanou de seus corações generosos, cada momento moldou quem sou hoje. Agradeço por cada sacrifício, por cada sorriso, por cada lágrima enxugada, e por cada abraço que foi mais do que palavras poderiam expressar. Vocês são minha fonte de inspiração, meu refúgio seguro, e a razão pela qual sempre busco ser a melhor versão de mim mesmo. Este agradecimento é um tributo singelo à grandiosidade do amor e apoio que tenho recebido de vocês ao longo dos anos. Meu coração transborda de gratidão por tudo que fizeram e continuam fazendo por mim. Amo vocês mais do que as palavras podem capturar.

"Mudamos o mundo todos os dias. Mas para mudar o mundo de um jeito significativo leva muito mais tempo do que as pessoas têm. Nunca acontece ao mesmo tempo. É devagar. É metódico. É exaustivo." (Mr. Robot)

RESUMO

Atualmente, a Internet das Coisas (IoT), um paradigma que conecta objetos à Internet sem intervenção humana, cresce e se integra cada vez mais às nossas vidas. No entanto, a segurança da IoT torna-se uma preocupação crescente devido aos seus recursos limitados (memória, energia e armazenamento), tornando-a um alvo potencial para diversos ataques, sendo o Distributed Denial of Service (DDoS) um dos mais comuns. Este tipo de ataque interrompe os serviços da IoT, prejudica o acesso de usuários legítimos e torna o serviço indisponível, podendo resultar em consequências desastrosas. Diante desse cenário, é crucial desenvolver soluções de proteção para a IoT. A Rede Definida por Software (SDN), que separa os planos de dados e de controle, proporcionando controle centralizado e visão global da rede, emerge como uma abordagem atrativa para fortalecer a segurança da IoT. Além disso, o uso do Parâmetro de Hurst, associado à autossimilaridade, possibilita a detecção eficiente de ataques DDoS, sendo um método leve ideal para ambientes de IoT. Neste trabalho, apresentamos o DDoS Detection System based on Hurst Parameter (DDSHP), um sistema que utiliza a SDN na IoT e o cálculo de Hurst do tráfego para detectar ataques de negação de serviço. Experimentos demonstram a eficiência desse sistema em redes IoT pequenas, como casas inteligentes, evidenciando também um tempo de resposta rápido aos ataques.

Palavras-chave: internet das coisas; segurança; rede definida por software; parâmetro de hurst; negação de serviço distribuída.

ABSTRACT

Currently, the Internet of Things (IoT), a paradigm that connects objects to the Internet without human intervention, is constantly growing and increasingly becoming a part of our lives. However, a growing concern with IoT is its security. After all, due to its limited resources (memory, energy, and storage), an IoT network becomes a potential target for various attacks, with Distributed Denial of Service (DDoS) being one of the most common. This attack disrupts IoT services, prevents access by legitimate users, and renders the service unavailable. This unavailability can lead to disastrous consequences. In this scenario, it is necessary to develop solutions that protect IoT from this type of attack. Software-Defined Networking (SDN) is a paradigm that separates data and control planes, ensuring centralized control and a global view of a network. As such, it becomes an attractive concept for securing IoT. Furthermore, with the Hurst Parameter, commonly linked to self-similarity, it is possible to detect DDoS attacks, making it an ideal method for an IoT environment due to its lightweight nature. Therefore, in this work, we propose and present a system called DDoS Detection System based on Hurst Parameter (DDSHP), which is an effort aimed at protecting the IoT network from DDoS attacks by applying SDN to IoT and using the Hurst parameter calculation to detect denial-of-service attacks. Through experiments, we show that this system is capable of detecting denial-of-service attacks with high efficiency in small IoT networks, such as smart homes, and exhibits a quick response time to attacks.

Keywords: internet of things; security; software defined network; hurst parameter; distributed denial of service.

LISTA DE FIGURAS

Figura 1 – Arquitetura IoT de 3 camadas. 22
Figura 2 – Aplicações da Internet das Coisas (IoT). 23
Figura 3 – Arquitetura SDN. 29
Figura 4 – Captura de junho de 2021 do tráfego de uma rede residencial de Crateús em 4 escalas de tempo diferentes: (a) 1ms - (b) 10ms - (c) 100ms - (d) 1s. . . . 37
Figura 5 – Infraestrutura do DDSHP. 49
Figura 6 – Fluxo de trabalho do DDSHP. 52
Figura 7 – Arquitetura dos cenários. 55
Figura 8 – Resultados do parâmetro de Hurst com nível de confiança de 99% para cada tráfego - Cenário *Smart Home* (20 sensores). 60
Figura 9 – Resultados do parâmetro de Hurst com nível de confiança de 99% para cada tráfego - Cenário *Smart Market* (50 sensores). 61
Figura 10 – Resultados do parâmetro de Hurst com nível de confiança de 99% para cada tráfego - Cenário *Smart Hospital* (100 sensores). 62
Figura 11 – Matriz de confusão para detecção de ataques. 63
Figura 12 – Resultados do tempo de detecção. 68

LISTA DE TABELAS

Tabela 1 – Impacto do DDoS em aplicações IoT 26
Tabela 2 – Classificação dos mecanismos e estratégias que empregam SDN para a segurança IoT. 36
Tabela 3 – Etapas de configuração do DDSHP. 51
Tabela 4 – Padrões de tráfego dos sensores dos cenários. 57
Tabela 5 – Composição dos sensores de cada cenário. 58
Tabela 6 – Resultados do Experimeto 2. 65

LISTA DE ABREVIATURAS E SIGLAS

6LoWPAN	*IPv6 over Low power Wireless Personal Area Networks*
API	*Application Programming Interface*
ATM	*Asynchronous Transfer Mode*
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CM	*Collector Module*
DDoS	Negação de Serviço Distribuída
DDSHP	*DDoS Detection System based on Hurst Parameter*
DM	*Detection Module*
DoS	Negação de Serviço
FN	Falso Negativo
FP	Falso Positivo
GB	gigabytes
H	Expoente de Hurst
ICMP	*Internet Control Message Protocol*
IDS	Sistema de Detecção de Intrusão
IIoT	Internet Industrial das Coisas
IoT	Internet das Coisas
IoV	Internet de Veículos
IP	*Internet Protocol*
LEDEM	*Learning-driven Detection Mitigation Mechanism*
LRD	Dependência de Longo Alcance
MAC	*Media Acess Control*
ML	Aprendizado de Máquina
MQTT	*Message Queuing Telemetry Transport*
MTD	*Moving Target Defense*
ONF	*Open Networking Foundation*
RAM	*Random Access Memory*
RFID	Identificação por Radiofrequência
RSSF	Rede de Sensores Sem Fio
SD-IoT	IoT Definida por *Software*

SDN	Rede Definida por Software
SECOD	*SDN Secure Control and Data Plane*
SRD	Dependência de Curto Alcance
SYN	*Synchronize*
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
WiFi	*Wireless Fidelity*

SUMÁRIO

1 INTRODUÇÃO 15
1.1 Objetivos 17
1.1.1 Objetivo Geral 18
1.1.2 Objetivos Específicos 18
1.2 Organização do Trabalho 18
2 FUNDAMENTAÇÃO TEÓRICA 20
2.1 Internet das Coisas (IoT) 20
2.1.1 Segurança em IoT 24
2.2 Software-Defined Networking (SDN) 27
2.2.1 SDN e Segurança em IoT 32
2.3 Parâmetro de Hurst (Autossimilaridade) 36
3 TRABALHOS RELACIONADOS 40
3.1 Parâmetro de Hurst na detecção de DDoS 40
3.2 SDN na proteção IoT contra DDoS 42
4 METODOLOGIA 46
5 PROPOSTA 48
5.1 Infraestrutura 49
5.2 Configuração do Sistema 50
5.3 Funcionamento 52
6 DESENVOLVIMENTO 54
6.1 Escopo 54
6.2 Recursos 57
6.3 Resultados 58
6.3.1 Experimento 1 (Estudo de Hurst) 59
6.3.1.1 Considerações *62*
6.3.2 Experimento 2 (Matriz de Confusão) 63
6.3.2.1 Considerações *66*
6.3.3 Experimento 3 (Tempo de Detecção) 67
6.3.3.1 Considerações *68*
7 CONCLUSÕES E TRABALHOS FUTUROS 69

REFERÊNCIAS 72

1 INTRODUÇÃO

A IoT é um paradigma em ascensão que está crescendo rapidamente. Segundo o relatório da IHS Markit (2018), estima-se que o número de dispositivos IoT possa atingir a marca de 73 bilhões até o ano de 2025. Esses dispositivos são essencialmente objetos físicos que podem se comunicar, trocar informações e tomar decisões de forma autônoma, sem a necessidade de intervenção humana (AL-FUQAHA *et al.*, 2015). Essa capacidade abre espaço para uma ampla variedade de aplicações em diversos setores, incluindo saúde, transporte, gerenciamento de energia, logística e automação residencial (ATZORI *et al.*, 2010). Na essência, a IoT se refere a uma rede que conecta objetos cotidianos à *Internet*. Esses objetos podem ser desde televisores, lâmpadas, câmeras e sensores, até *smartphones*.

O crescimento da IoT implica em uma crescente interconexão global, já que sua ideia fundamental é possibilitar a comunicação autônoma e a troca de dados entre dispositivos e aplicativos do mundo real (FAN; CHEN, 2010). No entanto, apesar de estar revolucionando o mundo atual, a IoT enfrenta alguns desafios significativos, com a segurança emergindo como um dos principais deles dentro deste paradigma (ATZORI *et al.*, 2010). Isso se deve, em parte, à limitação de recursos nos dispositivos IoT, tais como memória, armazenamento e capacidade de processamento, o que torna desafiador a implementação de soluções de segurança que demandam alto poder computacional (RAFIQUE *et al.*, 2019).

Devido às suas limitações intrínsecas e ao seu ambiente amplamente distribuído, a IoT se torna um alvo vulnerável para uma variedade de ataques, sendo um dos mais comuns o Negação de Serviço (DoS). Esse tipo de ataque representa uma ameaça significativa para os sistemas IoT, pois pode comprometer os canais de comunicação e sobrecarregar as redes IoT com um volume massivo de dados, exaurindo rapidamente os recursos disponíveis e resultando na indisponibilidade da rede (FARRIS *et al.*, 2019). Além disso, o problema se agrava quando o ataque ocorre de maneira distribuída, conhecida como Negação de Serviço Distribuída (DDoS), amplificando ainda mais seu impacto.

Os ataques de negação de serviços são um dos mais comuns na IoT (PACHECO *et al.*, 2016) e com grade variedade (*Synchronize* (SYN) *Flood*, Ping da Morte, *Internet Control Message Protocol* (ICMP) *Flood*, etc.). Para ilustrar as consequências de um ataque, consideremos o cenário de uma cidade inteligente em que todo o controle de tráfego é gerenciado por dispositivos autônomos. Se um ataque de DDoS tornar o serviço indisponível, isso poderia resultar em acidentes de trânsito como uma possível consequência direta. Isso destaca a gravidade da

interrupção de serviços em sistemas e redes IoT.

O processo de proteção da IoT continua em seus estágios iniciais de desenvolvimento, resultando em um escasso trabalho direcionado à defesa contra ataques de DDoS em ambientes de IoT (WANI; REVATHI, 2020). Além disso, as restrições e a grande heterogeneidade inerentes à IoT tornam as soluções de segurança convencionais inadequadas para este ambiente, exigindo a concepção de novas estratégias de proteção capazes de garantir a segurança de forma escalável e eficaz (YU *et al.*, 2015).

A Rede Definida por Software (SDN) oferece oportunidades promissoras para abordar questões relacionadas à segurança na IoT (KANAGAVELU; AUNG, 2019). Esse paradigma reconfigura a estrutura tradicional das redes, separando o plano de controle do plano de dados. Isso significa que os *switches* e roteadores se tornam dispositivos de encaminhamento simples, enquanto a lógica de controle é centralizada em um controlador SDN, conhecido como controlador, que mantém uma visão global da rede (KREUTZ *et al.*, 2015). Portanto, devido à sua abordagem centralizada, a SDN tem a capacidade de monitorar e gerenciar integralmente toda a rede por meio desse controlador.

Assim sendo, a SDN tem o potencial de introduzir novos mecanismos de segurança na IoT, permitindo que a rede responda de forma ágil a ataques suspeitos, identificando, bloqueando ou redirecionando ações suspeitas para proteger tanto os usuários quanto a própria rede (KANAGAVELU; AUNG, 2019). No contexto de ataques de DDoS, as soluções que fazem uso dos recursos proporcionados pela arquitetura SDN têm demonstrado eficácia na detecção e mitigação desses ataques, graças aos benefícios da programabilidade introduzida por meio de técnicas de *softwarization* (VISHWAKARMA; JAIN, 2020). Isso ilustra como a SDN pode ser um caminho viável para a proteção da IoT, desde que suas características intrínsecas sejam exploradas de maneira apropriada.

Conforme mencionado, a SDN demonstrou ser eficaz na detecção de ataques de DDoS. No âmbito da detecção de DDoS, alguns estudos têm adotado a combinação da SDN com técnicas de Aprendizado de Máquina (ML) para essa finalidade (alguns desses trabalhos são citados no Capítulo 3). Contudo, um método notável é a detecção baseada em estatísticas, que envolve a coleta e análise de amostras de dados da rede para identificar tráfego malicioso, aplicando algoritmos estatísticos desenvolvidos com diversas medidas (GALEANO-BRAJONES *et al.*, 2020). Entre as medidas utilizadas nesse método, destaca-se o uso da autossimilaridade do tráfego.

A autossimilaridade é uma propriedade fractal do tráfego de rede descoberta por Leland *et al.* (1994). Essa característica faz com que o tráfego de rede pareça qualitativamente o mesmo em escalas de tempo suficientemente grandes e exibe uma Dependência de Longo Alcance (LRD) (BARSUKOV *et al.*, 2019). Em outras palavras, independentemente da escala de tempo considerada, o tráfego mantém semelhanças consigo mesmo. Essa propriedade de autossimilaridade já foi identificada em alguns trabalhos relacionados à IoT (KOTENKO *et al.*, 2020).

Devido à sua característica de LRD, a autossimilaridade pode ser quantificada pelo Expoente de Hurst (H), utilizado para medir numericamente a extensão da LRD. Quando a propriedade de LRD é acentuada, a autossimilaridade também é forte, resultando em um valor do H entre 0,5 e 1. Em contrapartida, quando a LRD é fraca, a autossimilaridade não está presente (YU *et al.*, 2016).

Uma das consequências significativas da autossimilaridade no tráfego é a capacidade de detectar ataques e outras anomalias de rede por meio da análise fractal desse tráfego (BARSUKOV *et al.*, 2019). Estudos, como o realizado por Deka e Bhattacharyya (2016), demonstraram que ataques de DDoS resulta em uma modificação no valor do H, indicando a ocorrência de problemas na rede. Essa observação sugere que a análise de Hurst pode ser uma ferramenta para a detecção precoce de ataques e anomalias na IoT e em outros ambientes de rede.

Com base no exposto, este documento apresenta a proposta de um sistema denominado *DDoS Detection System based on Hurst Parameter* (DDSHP), destinado a proteger as redes IoT contra ataques de DDoS. Este sistema utiliza a tecnologia SDN para auxiliar na detecção de ataques de DDoS em um ambiente IoT. O DDSHP realiza a detecção de ataques calculando Hurst, identificando assim a ocorrência de um ataque pela alteração de seu valor no tráfego.

1.1 Objetivos

Diante do problema exposto e das possíveis abordagens para intervenção, esta seção destaca os objetivos fundamentais que nortearam a condução da pesquisa. O delineamento claro desses objetivos não apenas proporciona uma visão mais precisa sobre os propósitos do trabalho, mas também estabelece as bases para a metodologia adotada. A compreensão desses objetivos é essencial para contextualizar a relevância do estudo e a contribuição que se busca proporcionar ao campo em questão.

1.1.1 Objetivo Geral

O propósito central desta pesquisa consiste no desenvolvimento de um sistema eficaz para salvaguardar redes IoT contra ataques de DDoS. A abordagem adotada envolve a utilização do valor de Hurst como ferramenta na identificação desses ataques, em conjunto com a implementação da SDN para a coleta de estatísticas de tráfego essenciais ao processo de cálculo. Este enfoque integrado visa fortalecer a segurança da rede IoT.

1.1.2 Objetivos Específicos

A partir do delineamento do objetivo geral, desdobram-se os objetivos específicos. Estes atuam como guias, direcionando o foco e proporcionando uma abordagem mais detalhada. Neste contexto, os objetivos específicos do trabalho são:

a) Investigar o comportamento do valor de Hurst, comumente ligado a autossimilaridade, no tráfego de uma rede IoT;
b) Observar que o cálculo de Hurst pode ser usado para identificar a ocorrência de ataques de DDoS em redes IoT;
c) Identificar como a SDN pode ser útil em redes IoT;
d) Definir os módulos que compõem o sistema desenvolvido;
e) Exibir a rápida detecção de ataques de negação de serviço (DoS e DDoS) nas redes IoT usando Hurst.

1.2 Organização do Trabalho

Este trabalho é composto por sete capítulos, sendo esta introdução o ponto de partida que contextualiza o trabalho, explicitando seus objetivos.

No Capítulo 2, abordamos os temas fundamentais deste trabalho: IoT, SDN e o Parâmetro de Hurst. Apresentamos os conceitos essenciais para compreender o escopo da pesquisa, discutindo as implicações de segurança na IoT. Destacamos o papel da SDN e do Parâmetro de Hurst nesse contexto, com ênfase especial na segurança relacionada a ataques de DDoS.

No Capítulo 3, apresentamos os trabalhos relevantes na literatura relacionados aos objetivos da pesquisa. Dividimos esta seção em duas partes: a primeira aborda estudos que utilizaram o Parâmetro de Hurst para detectar ataques de DDoS, enquanto a segunda parte se

concentra em pesquisas que empregaram a SDN para proteger as redes IoT contra ataques de DDoS.

No Capítulo 4, abordamos as escolhas e estratégias adotadas na condução da pesquisa, proporcionando uma visão clara do desenvolvimento do presente projeto.

No Capítulo 5, delineamos o DDSHP, expondo detalhadamente sua infraestrutura, configuração e o modo como opera. Aqui exploramos cada componente e proporcionamos uma visão aprofundada que se torna fundamental para o entendimento pleno do sistema proposto.

No Capítulo 6, abordamos o desenvolvimento da proposta e exploramos os experimentos realizados com o objetivo principal de analisar os aspectos relacionados ao DDSHP. Além disso, analisamos os resultados dos experimentos realizados.

No decorrer do Capítulo 7, destacam-se as conclusões alcançadas ao longo deste trabalho, proporcionando uma síntese dos principais resultados obtidos. Além disso, delineamos as diretrizes futuras que se apresentam como próximos passos naturais a serem tomados no desenvolvimento da pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, abordamos os temas fundamentais deste trabalho, que incluem IoT, SDN e o Parâmetro de Hurst. Apresentamos os principais conceitos necessários para compreender o escopo geral da pesquisa. Além disso, discutimos as implicações de segurança na IoT e como a SDN e o Parâmetro de Hurst podem desempenhar um papel nesse contexto, com um foco especial na segurança relacionada a ataques de DDoS.

2.1 Internet das Coisas (IoT)

A ideia central desse conceito é a onipresença, em nosso dia-a-dia, de uma variedade de coisas ou objetos que, por meio de esquemas de endereçamento únicos, têm a capacidade de interagir uns com os outros e colaborar para atingir objetivos compartilhados (ATZORI *et al.*, 2010). Dessa forma, podemos compreender que a IoT conecta objetos do mundo real e incorpora inteligência ao sistema para processar informações específicas desses objetos e tomar decisões autônomas que sejam úteis (HUANG; LI, 2010). Isso pode ser exemplificado pela iluminação que se acende automaticamente ao detectar movimento ou pelo ar-condicionado que ajusta a temperatura com base nas informações ambientais em tempo real.

Como observado, a IoT possibilita que objetos físicos cotidianos interajam virtualmente, permitindo que estejam cientes de eventos que ocorrem a grandes distâncias ou respondam a eventos que não podem ser detectados fisicamente (HAYAJNEH *et al.*, 2020). Assim, uma variedade de dispositivos, como câmeras de segurança, alarmes, sensores de temperatura, movimento e fumaça, bem como objetos de uso diário com acesso à *Internet*, como televisores, geladeiras, *smartphones* e carros, são considerados parte do domínio da IoT. Indiscutivelmente, a principal força da ideia da IoT reside no alto impacto que ela terá em diversos aspectos da vida cotidiana e no comportamento dos potenciais usuários (ATZORI *et al.*, 2010).

Muitos conceitos e tecnologias contribuíram para o surgimento da IoT. Um deles é a Rede de Sensores Sem Fio (RSSF), que consiste em vários nós sensores capazes de coletar, processar e transmitir dados em diversos ambientes por meio de comunicação sem fio (GUBBI *et al.*, 2013). Outra tecnologia fundamental é a Identificação por Radiofrequência (RFID), um sistema de baixo custo e tamanho reduzido, que opera independentemente da energia da bateria, composto por *tags*, leitores e um servidor *backend* (GUBBI *et al.*, 2013). Esses exemplos destacam a amplitude da IoT e como sua infraestrutura pode ser construída a partir de redes

existentes e em constante evolução.

De acordo com Miorandi *et al.* (2012), os dispositivos inteligentes na IoT têm características físicas bem definidas, muitas vezes restritas em termos de capacidade computacional. Geralmente, esses dispositivos apresentam limitações como memória limitada, poder de processamento limitado e uma fonte de energia limitada, que pode ser escassa ou inexistente. Além disso, os mecanismos de comunicação utilizados por alguns dispositivos IoT, que frequentemente operam em redes sem fio, podem ter potência de transmissão reduzida e baixas taxas de transferência de dados. No entanto, é importante destacar que essas limitações não impedem a formação de redes compostas por milhões de dispositivos IoT heterogêneos.

As diversas interações entre os dispositivos IoT resultam na geração de enormes volumes de dados e na oferta de uma ampla variedade de serviços, o que intensifica a necessidade de indexar, agregar, armazenar e processar esses dados de maneira mais eficiente (ITU, 2012). Além disso, é importante destacar que os dispositivos IoT geralmente se conectam e se comunicam apenas com outros dispositivos que implementam o mesmo serviço específico. Conforme descrito por Khan *et al.* (2012), o fluxo de trabalho básico simplificado da IoT pode ser resumido da seguinte maneira:

a) **Sensoriamento de objetos e comunicação de informações específicas:** Nesta etapa, os dispositivos IoT realizam a detecção e coleta de informações específicas, que podem incluir dados como temperatura, orientação, movimento, vibração, aceleração, umidade, mudanças químicas no ar, entre outros;
b) **Acionamento de ações automatizadas:** As informações capturadas pelos dispositivos IoT são processadas por um dispositivo central ou sistema, que toma decisões com base nesses dados e determina ações automatizadas a serem executadas em resposta às informações recebidas;
c) **Fornecimento de serviços e *feedback*:** Nesta etapa, os dispositivos IoT executam as ações determinadas anteriormente e fornecem serviços com base nas informações coletadas. Além disso, eles incluem um mecanismo para fornecer *feedback* ao administrador do sistema, informando sobre o status atual do sistema e os resultados das ações realizadas.

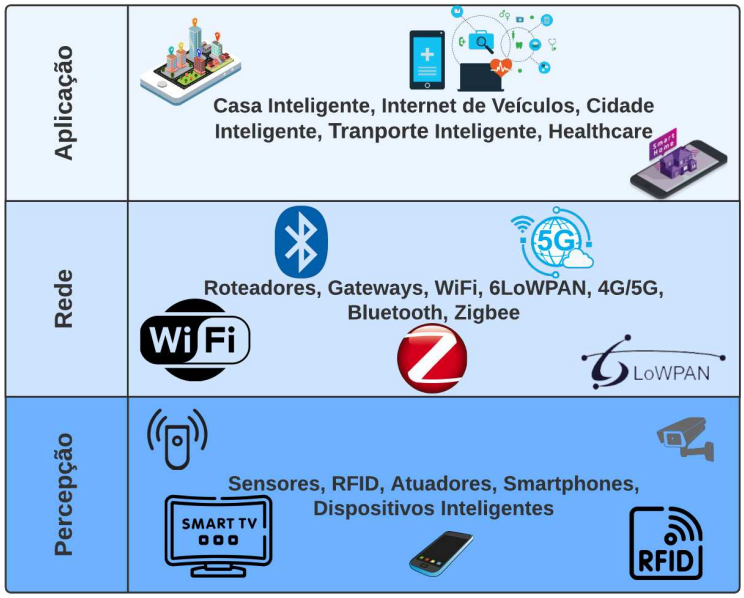
Portanto, a arquitetura da IoT desempenha um papel crucial ao garantir que esse fluxo de trabalho ocorra eficientemente, preenchendo a lacuna entre os mundos físico e virtual. Para alcançar esse objetivo, é essencial adotar uma arquitetura em camadas que seja flexível e

leve em consideração uma série de fatores, especialmente a limitação dos dispositivos IoT e sua heterogeneidade. No entanto, é importante notar que não existe um consenso único sobre uma arquitetura universalmente aceita para a IoT (SETHI; SARANGI, 2017).

Algumas propostas de arquitetura defendem um modelo de cinco camadas, como visto em trabalhos como Khan *et al.* (2012) e Al-Fuqaha *et al.* (2015). No entanto, uma abordagem mais simplificada de três camadas, que encapsula a ideia principal da IoT, tem sido amplamente adotada, como evidenciado em estudos como Abbou *et al.* (2018), Kanagavelu e Aung (2019). A escolha entre diferentes arquiteturas dependerá das necessidades específicas do projeto e das características dos dispositivos IoT envolvidos.

Em uma arquitetura de três camadas, conforme ilustrado na Figura 1, cada camada desempenha um papel específico:

Figura 1 – Arquitetura IoT de 3 camadas.



Fonte: Adaptado de Sethi e Sarangi (2017).

a) **Camada de Percepção:** A camada de percepção tem a função de coletar, processar e interpretar as informações provenientes do mundo físico (KANAGAVELU; AUNG, 2019). Ela é composta por dispositivos RFID, sensores e atuadores de diversos tipos. Essa camada é responsável por capturar dados como temperatura, umidade, movimento, entre outros, dos objetos físicos;

b) **Camada de Rede:** Também conhecida como camada de transmissão, a camada de rede é responsável por transmitir os dados coletados pelos objetos físicos para o processamento (ABBOU *et al.*, 2018). Essa transmissão pode ocorrer por meio

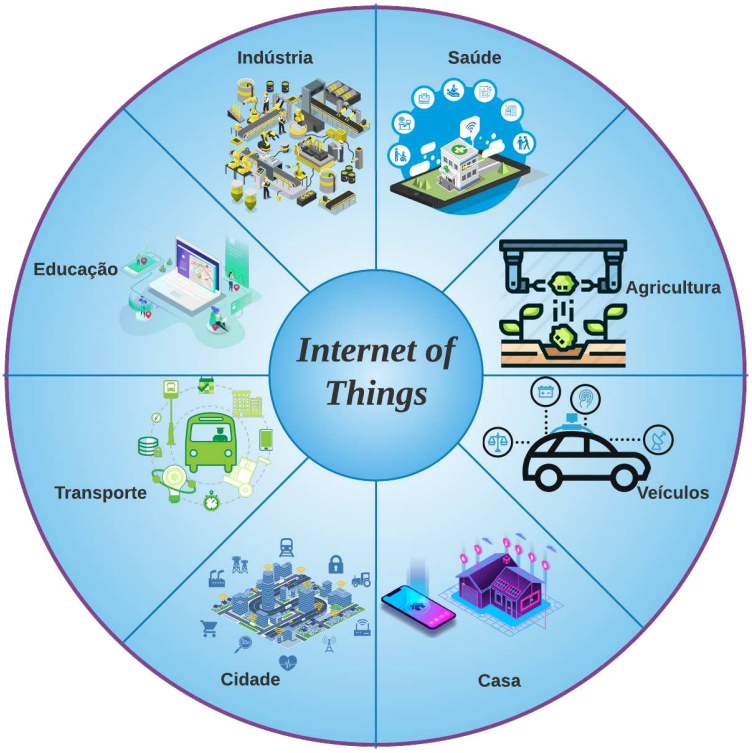
de diversos meios, como *Wireless Fidelity* (WiFi), 4G/5G, Bluetooth, ZigBee, tanto com fio quanto sem fio;

c) **Camada de Aplicação**: A camada de aplicação, também chamada de camada de processamento, é responsável por resolver os desafios relacionados ao processamento de informações e à interface entre o usuário e o sistema (Lei Zhang *et al.*, 2014). Nesta camada, os dados provenientes da camada de rede são processados e utilizados para dar suporte a diversas categorias de aplicações IoT existentes, como Casa Inteligente, Cidade Inteligente, Internet de Veículos (IoV), *Healthcare* e outras.

Essa arquitetura de três camadas é fundamental para a operação eficiente da IoT, permitindo que os dados sejam coletados, transmitidos e processados para suportar uma ampla gama de aplicações e serviços.

Portanto, fica claro que a IoT se integrou de forma significativa em nossa vida cotidiana, com dispositivos IoT presentes em uma ampla variedade de áreas, incluindo residências, edifícios, cidades, fazendas, instituições educacionais e até mesmo em nossos próprios corpos (Figura 2).

Figura 2 – Aplicações da IoT.



Fonte: Adaptado de Miorandi *et al.* (2012).

Em virtude dessa diversidade de dispositivos IoT, a previsão é que o número total deles alcance a impressionante marca de 73 bilhões até 2025 (IHS Markit, 2018). Isso reforça a crescente importância e popularidade da IoT, que está destinada a desempenhar um papel ainda mais proeminente em um futuro próximo, consolidando-se como uma parte integral do mundo contemporâneo em diversos setores.

2.1.1 Segurança em IoT

A segurança é um dos principais desafios enfrentados pela IoT. A maioria dos componentes da IoT é caracterizada por recursos limitados, incluindo capacidades limitadas de energia e computação, o que torna impraticável a implementação de esquemas complexos de segurança (ATZORI *et al.*, 2010). Além disso, os dispositivos IoT permanecem geralmente conectados à internet (mesmo quando não há necessidade), carecem de proteção antivírus eficaz e possuem mecanismos de segurança fracos (BANERJEE; SAMANTARAY, 2019). Além disso, as soluções de segurança tradicionais, como *firewalls* e criptografia, muitas vezes não podem ser aplicadas com êxito em ambientes de IoT (TAHAEI *et al.*, 2020). Portanto, é compreensível que os dispositivos IoT sejam alvos atraentes para ataques cibernéticos, dado o cenário de vulnerabilidade que frequentemente apresentam.

Pesquisadores têm identificado uma série de desafios de segurança na IoT, que incluem questões relacionadas à segurança de rede, gerenciamento de identidade, privacidade, heterogeneidade, escalabilidade e disponibilidade (YOUSUF; MIR, 2020). Muitos desses desafios se concentram especialmente na questão da privacidade. Em redes heterogêneas, como as encontradas na IoT, garantir a privacidade dos usuários não é uma tarefa simples, uma vez que, devido à natureza da comunicação, as informações são frequentemente transmitidas sem confidencialidade e autenticidade adequadas (HAYAJNEH *et al.*, 2020). Essas preocupações destacam a necessidade de abordagens e soluções de segurança adaptadas às particularidades da IoT.

No entanto, a disponibilidade também é uma preocupação, pois os dados provenientes de sensores e outros dispositivos IoT devem estar prontamente acessíveis quando necessário (IOULIANOU *et al.*, 2018). A disponibilidade desempenha um papel importante nos serviços de IoT que possibilitam acesso de qualquer lugar e a qualquer momento (IDRIS; HAMEED, 2016). Isso é vital para serviços de IoT em tempo real que dependem da disponibilidade constante de dados, como os utilizados em setores como transporte, agricultura, indústria e saúde, que

fornecem informações de forma contínua. Portanto, garantir a disponibilidade dos dados é fundamental para o funcionamento eficaz desses serviços.

Atualmente, um dos principais ataques que afetam a disponibilidade de serviços é o DDoS, que é capaz de ameaçar a disponibilidade de um alvo em questão de segundos. Esse tipo de ataque é definido como uma tentativa de negar ou interromper o acesso legítimo a um serviço. O que torna o DDoS perigoso é que ele envolve o uso de múltiplos computadores e conexões à *internet* para criar uma ameaça real que pode bloquear ou prejudicar seriamente o acesso de outros usuários ao servidor hospedeiro (ZHANG; GREEN, 2015). A gravidade desse tipo de ataque já causou perdas financeiras substanciais para empresas e pode afetar milhões de usuários em todo o mundo (MARZANO *et al.*, 2018). Portanto, o ataque de DDoS é considerado uma das ameaças de segurança mais sérias para a IoT.

A IoT é suscetível a ataques de DDoS de duas maneiras distintas:

a) **Dispositivos IoT sendo infectados e utilizados como parte de ataques de DDoS:** Nesse cenário, os dispositivos IoT são comprometidos por invasores e transformados em parte de uma rede de *bots* (robôs) controlados remotamente. Essa rede de dispositivos comprometidos, também conhecida como *botnet*, é então utilizada para lançar ataques de DDoS em alvos específicos;
b) **Serviços e aplicações IoT como alvos de ataques de DDoS:** Nesse caso, os serviços e as aplicações IoT em si são alvos diretos de ataques de DDoS. Isso pode resultar em uma sobrecarga dos recursos de rede, tornando os serviços indisponíveis para os usuários legítimos e interrompendo as operações normais da IoT.

Ambas as situações representam sérias ameaças à segurança e à disponibilidade da IoT, destacando a importância de medidas de proteção adequadas contra ataques de DDoS.

A primeira forma de ataque refere-se à construção de uma *botnet*, que é uma rede de dispositivos IoT comprometidos que são facilmente dominados e controlados por hackers. Esses dispositivos são utilizados para gerar tráfego falso que, eventualmente, se reúne para formar um ataque DDoS (WANI; REVATHI, 2020).

De fato, estudos como Pa *et al.* (2015) identificaram que ataques de DDoS em grande escala são predominantemente causados por *botnets* IoT. A razão para essa preferência pelos dispositivos IoT como instrumentos para lançar ataques deve-se às suas limitações de recursos, à fraca proteção e à falta de supervisão, geralmente. Um exemplo notório desse tipo de ataque é

a *botnet Mirai*, que em 2016 interrompeu uma parte significativa da *internet*, afetando vários países, e utilizou muitos dispositivos IoT distribuídos geograficamente para orquestrar os ataques (SILVA *et al.*, 2020). Esse episódio ressalta o potencial impacto devastador dessas *botnets* IoT.

O DDoS também pode direcionar seus ataques às redes IoT, inundando-as com tráfego ilegítimo, o que resulta na indisponibilidade dos serviços. A Tabela 1 mostra como o DDoS pode afetar diferentes aplicações da IoT, sendo que o impacto mais significativo ocorre em aplicações de tempo real, como controle de tráfego e saúde. Nessas situações, a indisponibilidade pode causar atrasos, erros e perda de dados, o que pode ter consequências graves. Por exemplo, um ataque DDoS em uma rede IoT de um sistema de saúde pode colocar a vida dos pacientes em risco. Da mesma forma, um ataque a uma rede IoT veicular pode resultar em acidentes não controlados, representando uma ameaça para motoristas, passageiros e pedestres (WANI; REVATHI, 2020).

Tabela 1 – Impacto do DDoS em aplicações IoT

Aplicações IoT	**Impacto do DDoS**
Engenharia de tráfego Controle de rede elétrica Assistência médica	Alto
Sistemas de localização Agricultura Gestão industrial	Médio
Automação residencial Abastecimento de água Monitoramento do clima Controle de estacionamento	Baixo

Fonte: Silva *et al.* (2020).

Portanto, fica evidente que os ataques de DDoS direcionados às redes IoT representam uma ameaça significativa, com potencial para causar danos sérios em diversas áreas, incluindo saúde e segurança viária.

Outra ameaça associada aos ataques de DDoS é o esgotamento dos recursos da bateria e do dispositivo. Por exemplo, um adversário pode impedir que um dispositivo entre em modo de repouso enviando regularmente mensagens, ou pode sobrecarregar os recursos limitados de energia e memória do dispositivo ao enviar tarefas de computação intensiva (IOULIANOU *et al.*, 2018). Isso pode resultar em um consumo excessivo de energia, reduzindo a vida útil da bateria do dispositivo, além de torná-lo menos responsivo e menos eficiente em termos de

recursos. Portanto, os ataques de DDoS podem causar danos não apenas à disponibilidade dos serviços IoT, mas também ao desempenho e à eficiência dos próprios dispositivos.

Com base no que foi apresentado, fica claro que a diversidade e a dinâmica de uso da IoT criaram desafios em termos de segurança. Além disso, os ataques de DDoS estão se tornando um problema em constante crescimento, representando uma ameaça séria para os sistemas IoT. Esses ataques podem comprometer os *links* de comunicação e sobrecarregar as redes com volumes massivos de dados. Diante dessa situação, surge a necessidade de novas tecnologias que possam oferecer maneiras mais eficazes de proteger as redes IoT contra ataques de DDoS. Essa é uma questão crítica, dada a crescente dependência da sociedade em relação à IoT em várias áreas, desde a saúde até o transporte, tornando a segurança da IoT uma prioridade cada vez mais importante.

2.2 Software-Defined Networking (SDN)

As redes de computadores podem ser divididas em três planos de funcionalidades: dados (dispositivos de rede responsáveis por encaminhar os dados), controle (protocolos usados para preencher as tabelas de encaminhamento dos elementos do plano anterior) e gerenciamento (serviços de *software* usados para monitorar e configurar remotamente a funcionalidade de controle) (KREUTZ *et al.*, 2015). Dessa forma, vemos que cada plano tem suas funções e responsabilidades bem definidas.

Nas redes tradicionais, os planos de controle e de dados estão intimamente vinculados e incorporados nos mesmos dispositivos de rede, como *switches* e roteadores. Essa estrutura é altamente descentralizada e, embora seja eficaz, resultou em uma arquitetura complexa e relativamente inflexível (KREUTZ *et al.*, 2015). Como resultado, as redes tradicionais se tornaram difíceis de gerenciar, uma vez que a implantação de novas políticas exigia configurações separadas em cada dispositivo da rede.

Nesse contexto, a SDN surge como uma alternativa viável às redes tradicionais. Em sua definição original, a SDN se refere a uma arquitetura de rede em que o estado de encaminhamento no plano de dados é gerenciado de forma remota, separando assim o plano de dados do plano de controle (KREUTZ *et al.*, 2015). Isso significa que o encaminhamento dos dados é desacoplado fisicamente da lógica de controle da rede. Esse desacoplamento permite que uma SDN seja modificada de maneira rápida e fácil, conforme as necessidades em constante evolução (HAYAJNEH *et al.*, 2020).

Conforme evidenciado, o cerne do conceito de SDN reside na separação dos planos de dados e de controle. Essa separação possibilita a implementação de lógica de controle personalizada nos dispositivos físicos, segundo os requisitos específicos de cada aplicação em tempo real (BERA *et al.*, 2017). Essa flexibilidade é fundamental para adaptar a rede às demandas em constante mudança das aplicações e dos serviços.

Conforme descrito por Kreutz *et al.* (2015), a SDN se baseia em quatro pilares fundamentais:

a) **Desacoplamento dos Planos de Controle e Dados:** Isso implica na separação física entre os planos de controle e dados, onde a funcionalidade de controle é removida dos dispositivos de rede, que se tornam simples elementos de encaminhamento.;

b) **Decisões de Encaminhamento Baseadas no Fluxo:** Em vez de tomar decisões de encaminhamento com base no destino dos pacotes, a SDN utiliza informações sobre o fluxo, que representa a sequência de pacotes entre uma origem e um destino;

c) **Deslocamento da Lógica de Controle para uma Entidade Externa:** A lógica de controle é movida para uma entidade externa conhecida como controlador SDN. Esse controlador fornece os recursos e abstrações necessários para facilitar a programação dos dispositivos de encaminhamento de pacotes;

d) **Programabilidade por Aplicativos de *Software*:** A rede se torna programável por meio de aplicativos de software executados no controlador e interagem com os *switches*/roteadores do plano de dados. Isso permite a implementação de políticas de rede flexíveis e adaptáveis consoante as necessidades específicas da aplicação.

Esses quatro pilares formam a base da arquitetura SDN e são essenciais para a sua flexibilidade e capacidade de adaptação às demandas em constante mudança das redes modernas.

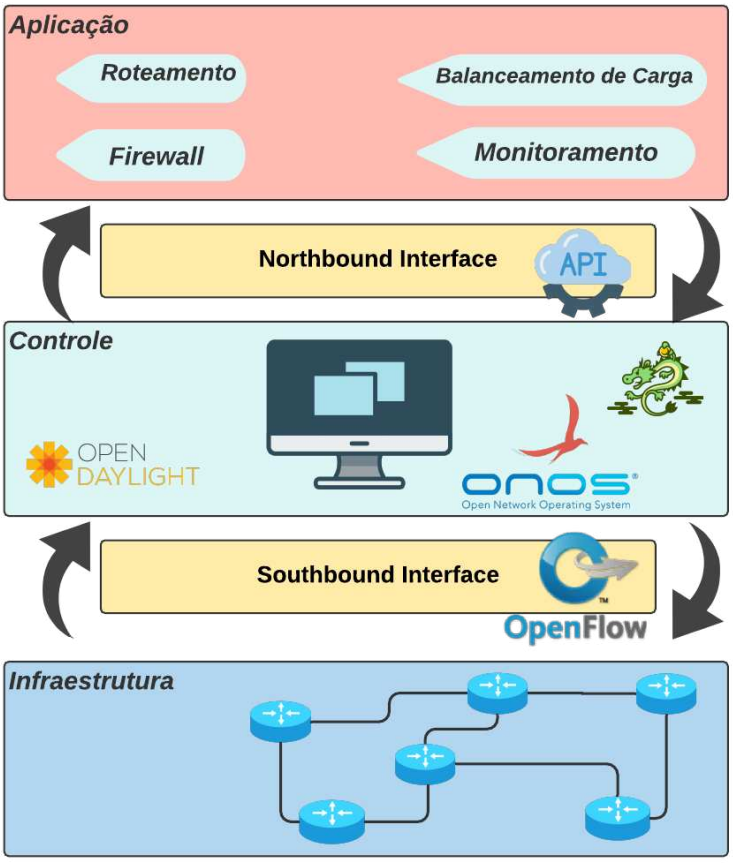
Como consequência desses pilares, a SDN traz consigo algumas vantagens como (KREUTZ *et al.*, 2015; MARTINEZ-JULIA; SKARMETA, 2014; TAYYABA *et al.*, 2017; DJOUANI *et al.*, 2018):

a) Torna-se mais fácil programar as aplicações de rede, pois as abstrações fornecidas pelo plano de controle e/ou as linguagens de programação de rede podem ser compartilhadas;

b) Todos os aplicativos da rede podem tirar proveito das mesmas informações (a visão de rede global), levando a decisões de política mais consistentes e eficazes;
c) Os aplicativos podem realizar ações (ou seja, reconfigurar dispositivos de encaminhamento) de qualquer parte da rede;
d) Definição de poderosos elementos de comutação (encaminhadores);
e) Facilita a alta transmissão de dados, eficiência espectral, alocação de recursos e gerenciamento de rede;
f) Flexibilidade para permitir a comunicação entre nós de redes heterogêneas;
g) Facilidade na integração de diferentes serviços de rede (balanceadores de carga, *firewalls*, sistemas de detecção de intrusão).

Segundo a *Open Networking Foundation* (ONF)(ONF, 2021), um consórcio sem fins lucrativos dedicado ao desenvolvimento, à padronização e à comercialização de SDN, o modelo de arquitetura de referência SDN é composto por três camadas (Figura 3).

Figura 3 – Arquitetura SDN.



Fonte: Elaborada pelo autor.

Nesse modelo, uma SDN consiste em três camadas principais: aplicação, controle e

infraestrutura. Essa estrutura foi desenvolvida com base nos planos de funcionalidades das redes de computadores, que incluem gerenciamento, controle e dados. Além disso, a infraestrutura da SDN possui duas interfaces, conhecidas como interfaces *Northbound* e *Southbound*, que facilitam a comunicação entre a camada de controle e as outras camadas. Abaixo, detalhamos cada camada e as interfaces:

a) **Camada de Aplicação**: essa é a camada de mais alto nível na arquitetura SDN com diferentes funções para atender às necessidades dos usuários finais, usando a interface *Northbound* para se comunicar com a de controle. Sendo assim, é a camada que contém o plano de gerenciamento. Ela é composta pelo conjunto de aplicativos que potencializam as funções oferecidas pela interface *Northbound* para implementar o controle de rede e a lógica de operação. (KREUTZ *et al.*, 2015). Dessa forma, é nesta camada em que os diferentes serviços de rede são oferecidos.

b) **Interface *Northbound***: é uma ponte entre a camada de aplicação e a de controle. Ela admite o desenvolvimento de aplicativos que abstrai os conjuntos de instruções de baixo nível usados pela interface *Southbound* para programar os dispositivos de encaminhamento (KREUTZ *et al.*, 2015), criando assim, uma visão abstraída de toda a rede para a camada de aplicação. Dessa forma, essa interface permite que os desenvolvedores de aplicativos controlem e programem a rede (SHIRALI-SHAHREZA; GANJALI, 2013). A *Northbound* é definida como um sistema de *software*, não de *hardware*, e que ainda não há um padrão estabelecido como na interface *Southbound* (KREUTZ *et al.*, 2015).

c) **Camada de Controle**: é a parte central da SDN e usa controladores para fornecer a funcionalidade de controle centralizado logicamente que supervisiona o comportamento de encaminhamento da rede (LI *et al.*, 2015). O controlador funciona como o cérebro da rede tendo uma visão global sobre ela e definindo regras de comunicação nos encaminhadores (KREUTZ *et al.*, 2015). Um controlador é um elemento crítico em uma arquitetura SDN, pois faz a ponte entre o plano de aplicativo (camada de aplicação) e o plano de dados (camada de infraestrutura), traduzindo os requisitos dos aplicativos em regras de encaminhamento apropriadas a serem aplicadas pelos comutadores de rede subjacente (FARRIS *et al.*, 2019). Dessa forma, a camada de controle SDN gerencia a

camada de infraestrutura e implementa políticas através da interface *Southbound*, além de fornecer uma visão global da rede subjacente para a de aplicação através da interface *Northbound* (SHIRALI-SHAHREZA; GANJALI, 2013). Além disso, inclui componentes básicos como gerenciador de dispositivos, unidade de processamento de pacotes, gerenciador de topologia e roteamento (DILLON; BERKELAAR, 2014);

d) **Interface *Southbound***: as interfaces *Southbound* são as pontes de conexão entre os elementos de controle e encaminhamento, sendo, portanto, o instrumento crucial para separar o controle e a funcionalidade do plano de dados (KREUTZ *et al.*, 2015). Ou seja, essa interface é responsável por realizar a comunicação entre a camada de controle e os dispositivos da camada de infraestrutura. O protocolo mais comum usado na interface *Southbound* é o OpenFlow, um protocolo importante no escopo SDN, mantido pela ONF e suportado por todos os principais fornecedores de equipamentos de rede (GALEANO-BRAJONES *et al.*, 2020). Este protocolo permite que os controladores de rede determinem os caminhos do fluxo através de uma rede de *switches*, permitindo assim o gerenciamento fácil do tráfego (VILALTA *et al.*, 2016). Além disso, fornece um controle detalhado para responder às mudanças em tempo real nos níveis de aplicativo, usuário e sessão em uma rede (KARAARSLAN *et al.*, 2020);

e) **Camada de Infraestrutura**: composta de um conjunto de equipamentos de rede (*switches*, roteadores e dispositivos *middlebox*) sendo simples elementos de encaminhamento, sem controle embutido ou *software* para tomar decisões autônomas (KREUTZ *et al.*, 2015). Esta camada representa o plano de dados onde os dispositivos de encaminhamento são interconectados através de meio sem fio ou com fio. Além disso, os dispositivos são explorados para processar pacotes com base nas regras fornecidas pelo controlador SDN (comunicando-se pela *Southbound*) e para coletar informações de *status* de rede, como topologia de rede e estatísticas de tráfego (FARRIS *et al.*, 2019).

Como observado anteriormente, a SDN oferece controle e gerenciamento automáticos e dinâmicos para uma ampla variedade de dispositivos de rede, serviços, topologias, rotas de tráfego e políticas de qualidade de serviço, tudo isso usando linguagens de alto nível e *Application Programming Interface* (API) (DJOUANI *et al.*, 2018). Portanto, essa abordagem se

mostra altamente adequada para ambientes tão dinâmicos quanto a IoT. A IoT, como um termo abrangente que engloba uma vasta gama de dispositivos, pode se beneficiar significativamente da SDN devido à sua notável flexibilidade e capacidade de programação (FARRIS *et al.*, 2019).

A integração desses paradigmas também simplifica a análise de informações e os processos de tomada de decisão na IoT. Além disso, a SDN oferece diversas ferramentas de depuração que podem ser aplicadas no ambiente IoT, melhorando a capacidade da rede para coletar dados e facilitar o processo de depuração (KALKAN; ZEADALLY, 2018). Portanto, a combinação de SDN e IoT pode ser aplicada em diversos domínios, como transporte, cidades inteligentes ou residências conectadas.

2.2.1 SDN e Segurança em IoT

Conforme mencionado por Krishnan *et al.* (2018), os principais critérios para aplicar a SDN na rede IoT são: a capacidade de conectar e gerenciar com segurança centenas ou até milhares de dispositivos heterogêneos; uma arquitetura elástica e escalável para balancear de forma dinâmica as cargas de trabalho e programação para impor políticas e aplicativos personalizados.

Como mencionado na Seção 2.2, a SDN é um paradigma adequado para o ambiente da IoT. Isso se deve à natureza da SDN, que incorpora um controlador capaz de gerenciar e supervisionar toda a rede, fornecendo uma visão global da topologia e do estado em tempo real (FARRIS *et al.*, 2019). Consequentemente, a SDN pode simplificar a gestão das redes IoT, oferecendo uma representação clara dos recursos e dispositivos que compõem a rede. Além disso, a SDN pode ser empregada para fortalecer a segurança necessária na IoT, pois possui a adaptabilidade necessária para enfrentar ameaças e vulnerabilidades existentes (ABBOU *et al.*, 2018).

Desde que os recursos da SDN sejam explorados de forma eficiente, é possível superar os desafios de segurança na IoT. As inúmeras vantagens dos recursos da SDN que podem reforçar a segurança das redes na IoT são discutidas em detalhes em Fajar e Purboyo (2018) e Shin *et al.* (2016). Estas vantagens incluem:

a) a separação dos planos de controle e de dados faz com que os fluxos de rede sejam controlados de maneira mais eficiente e eficaz, o que permite distinguir dinamicamente os fluxos de rede malignos dos benéficos e assim separando-os;
b) a visão global e centralizada da rede permite que os usuários recebam todas as

informações de *status* da rede, deferindo a identificação rápida de atividades maliciosas, resultando, portanto, num tempo de resposta curto;

c) a programação de rede fornecida por meio do uso de API (*Northbound*) garante o desenvolvimento de aplicativos de segurança, permitindo que os dispositivos IoT possam ser programados, por exemplo, para conduzir uma inspeção detalhada de pacotes no tráfego;

d) a simplificação do plano de dados permite a implantação de novos módulos de segurança conforme as políticas de segurança desejadas, que ocorre sem alterar os demais módulos.

Essas vantagens desempenham um papel fundamental no desenvolvimento de mecanismos de segurança para a IoT utilizando a SDN. Conforme ressaltado por Kalkan e Zeadally (2018), na literatura, esses mecanismos podem ser categorizados em três grupos principais:

a) **Soluções Baseadas em Rede:** Esses modelos abordam a estrutura arquitetônica dos elementos da rede;

b) **Soluções Baseadas em Tráfego:** O foco principal está nos fluxos de tráfego, lidando com suas propriedades para determinar atividades maliciosas;

c) **Soluções Baseadas em Criptografia:** Concentram-se nas propriedades criptográficas do ambiente para fornecer segurança.

É notável como a aplicação da SDN na segurança da IoT é abrangente, oferecendo uma ampla gama de opções na criação de mecanismos de proteção.

Nesse contexto, foram estabelecidos diversos esquemas baseados em SDN para combater ataques de DDoS, que podem ter um impacto significativo no desempenho do sistema IoT (FARRIS *et al.*, 2019). Aproveitando as vantagens da SDN, é possível detectar e mitigar eficazmente fluxos maliciosos que contribuem para ataques de DDoS. Essas soluções geralmente se baseiam na análise de tráfego para a detecção e mitigação desses ataques.

Conforme destacado por Galeano-Brajones *et al.* (2020), existem três categorias de mecanismos que utilizam a SDN para defender as redes IoT contra ataques de DDoS:

a) **Baseados em Estatísticas:** Esses mecanismos envolvem a coleta e análise de amostras de dados da rede para identificar tráfego malicioso. Isso é alcançado por meio da aplicação de algoritmos estatísticos que são desenvolvidos utilizando várias medidas;

b) **Baseados em Aprendizado de Máquina:** Nessa abordagem, são utilizados

algoritmos tradicionais de aprendizado de máquina para analisar e classificar diferentes fluxos de tráfego, visando identificar ataques maliciosos de negação de serviço;

c) **Baseados em Aplicações Específicas:** Nessa categoria, são criados aplicativos específicos que se baseiam em diversos tópicos de pesquisa, como *blockchain*, para alertar outros usuários sobre a ocorrência de um ataque de negação de serviço.

Portanto, fica evidente como a SDN contribui para o desenvolvimento de técnicas e estratégias eficazes para lidar com os ataques de DDoS nas redes IoT.

Em sua pesquisa, como descrito em Silva *et al.* (2020), foram identificadas as principais estratégias para detectar e mitigar ataques de negação de serviço que fazem uso do paradigma da SDN. Além disso, o estudo demonstra como essas estratégias estão sendo aplicadas pelos pesquisadores em redes IoT, seja individualmente ou em combinação. As estratégias identificadas são as seguintes:

a) **Filtragem de fluxo:** essa estratégia considera os campos presentes nos cabeçalhos dos pacotes que chegam aos dispositivos para bloquear fluxos classificados como maliciosos, sendo uma das alternativas mais práticas de implementação e que depende da coleta de estatísticas e inspeção de pacotes pelo controlador (SILVA *et al.*, 2020);

b) ***Honeypots*:** esta estratégia envolve o uso de sistemas em ambientes isolados e monitorados que simulam as características de um alvo legítimo para que as informações possam ser coletadas para atualizar as políticas atuais de detecção e mitigação. Embora seja uma técnica tradicional de mitigação de ataques, ela pode ser usada em combinação com SDN, para auxiliar o controlador na coleta de informações sobre tráfego malicioso (YAN *et al.*, 2018);

c) **Limitação de taxa:** nessa estratégia, o controlador SDN pode definir um limite máximo para o volume de tráfego que pode ser processado pela rede sem ela ficar sobrecarregada. Se atingir o limite, a rede rejeitará todos os pacotes subsequentes (SILVA *et al.*, 2020);

d) ***Moving Target Defense* (MTD):** envolve o uso de técnicas para reconfigurar e atualizar dinâmica e continuamente as características de uma rede ou sistema com base em um conjunto de valores aleatórios para tentar evitar que os invasores

tornem o sistema alvo indisponível (MA *et al.*, 2015). Uma das principais técnicas utilizadas é a randomização de endereços *Internet Protocol* (IP) e *Media Acess Control* (MAC), o que dificulta a descoberta de informações sobre *hosts* e serviços da rede durante o processo e evita possíveis ataques de DDoS (SILVA *et al.*, 2020);

e) ***Traceback*:** usa as informações nos cabeçalhos dos pacotes para definir a origem real de um invasor. Graças aos benefícios fornecidos pela visão holística do plano de controle SDN é possível mitigar o ataque com base no rastreamento (CHEN *et al.*, 2020a);

f) ***Request Prioritization*:** nesta estratégia, é definido um valor de prioridade para o processamento dos fluxos que chegam à rede. A prioridade funciona atribuindo um valor de confiabilidade padrão aos *hosts* de origem para cada novo pacote de entrada que chega ao controlador SDN. O valor de confiabilidade é baseado no histórico de tráfego de cada *host* (SILVA *et al.*, 2020).

Sem dúvida, fica claro como a SDN desempenha um papel fundamental na segurança das redes IoT. A capacidade de programação e automação proporcionada pela SDN permite o desenvolvimento de mecanismos adaptáveis que podem reagir prontamente a comportamentos suspeitos. Essa integração entre SDN e IoT oferece uma abordagem atraente para a detecção e mitigação de ataques de negação de serviço.

Como evidenciado em trabalhos como Galeano-Brajones *et al.* (2020) e Silva *et al.* (2020), a combinação da SDN e IoT resultou na criação de uma ampla variedade de mecanismos, estratégias e técnicas. Essa pesquisa e desenvolvimento contínuos estão impulsionando a segurança das redes IoT e garantindo que elas possam enfrentar os desafios cada vez mais sofisticados apresentados pelos ataques de negação de serviço. Portanto, a integração da SDN e IoT desempenha um papel crucial na construção de sistemas mais seguros e resilientes para o futuro da conectividade IoT.

Na Tabela 2, apresentamos um resumo do conteúdo discutido nesta subseção. Este aborda a classificação dos mecanismos desenvolvidos para segurança IoT que se baseiam na SDN, abrangendo não apenas a defesa contra ataques de DDoS, mas também diversos outros aspectos relevantes. Além disso, destacamos as principais estratégias que fazem uso da SDN para prevenir ataques de DDoS.

Tabela 2 – Classificação dos mecanismos e estratégias que empregam SDN para a segurança IoT.

Mecanismos para segurança IoT (KALKAN; ZEADALLY, 2018)	Rede Tráfego Criptografia	Classificação para soluções que usam SDN e são desenvolvidas para segurança IoT em diferentes aspectos (privacidade, autenticidade, disponibilidade etc.)
Mecanismos para a defesa de DDoS em IoT (GALEANO-BRAJONES *et al.*, 2020)	Estatística Aprendizado de Máquina Aplicativos Específicos	Classificação definida segundo as soluções que usam SDN e foram criadas para contribuir na defesa da IoT contra DDoS, sendo principalmente baseadas em tráfego
Estratégias para detectar e mitigar DDoS (SILVA *et al.*, 2020)	Filtragem de Fluxo *Honeypots* Limitação de Taxa MTD *Traceback* *Request Prioritization*	Principais estratégias para detectar e mitigar ataques de negação de serviço que empregam o paradigma da SDN e que vem sendo aplicadas nas redes IoT no desenvolvimento de mecanismos

Fonte: Elaborada pelo autor.

2.3 Parâmetro de Hurst (Autossimilaridade)

A autossimilaridade é um conceito associado à invariância de escala, o que significa que as características permanecem consistentes em várias escalas diferentes. Em outras palavras, um objeto é considerado autossimilar quando suas partes, quando observadas em ampliações diferentes, mantêm semelhança com a forma do objeto em sua totalidade (KAUR *et al.*, 2020). Isso implica que o mesmo padrão se reproduz em diferentes escalas ao longo do tempo.

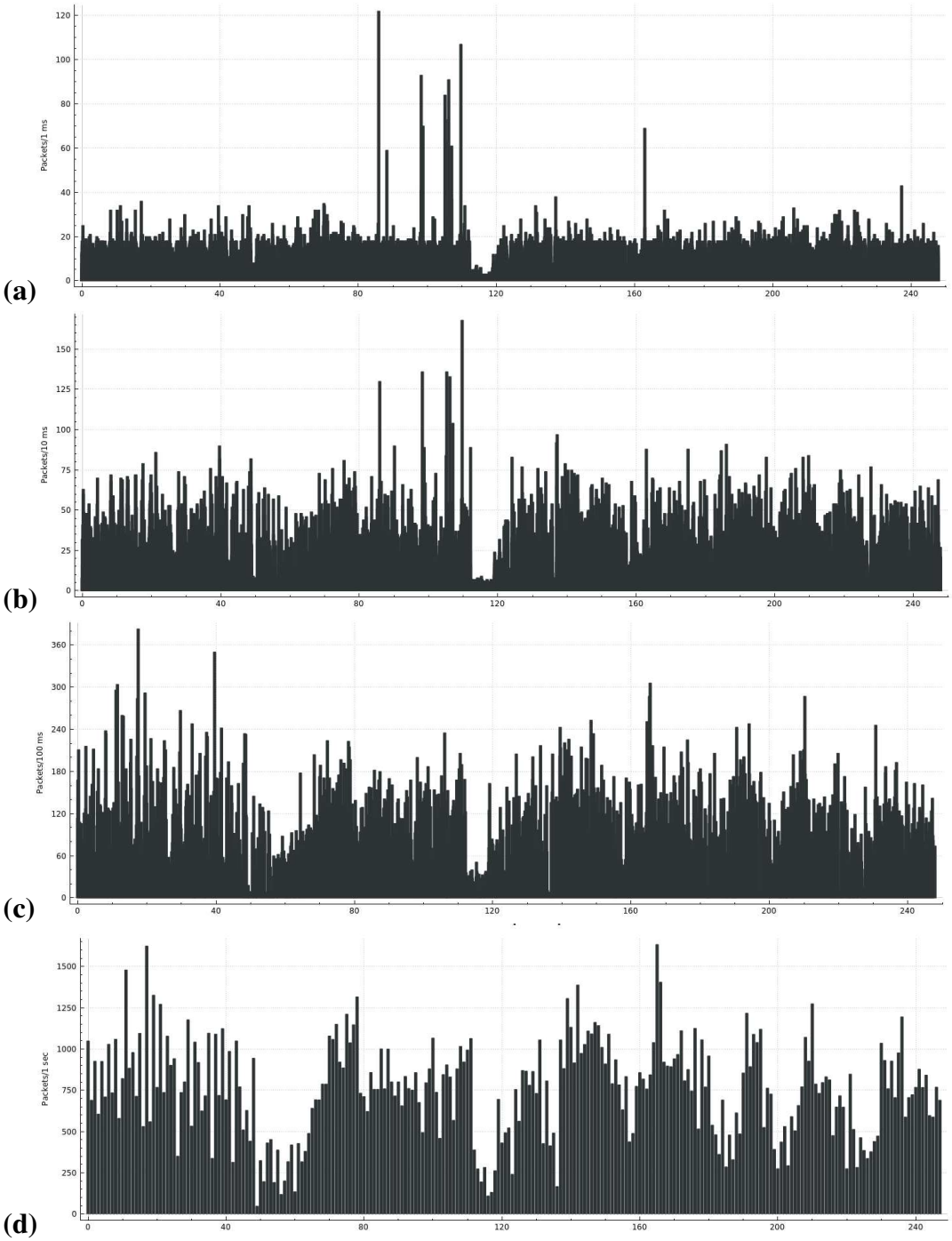
O pioneiro na identificação da autossimilaridade no tráfego foi o trabalho de Leland *et al.* (1994). Mediante uma análise estatística baseada em uma quantidade significativa de dados, essa pesquisa demonstrou que o tráfego exibe uma característica conhecida como LRD. A LRD é um fenômeno estatístico que se manifesta em processos autossimilares, como contraste aos processos de Dependência de Curto Alcance (SRD) adotados previamente. Estes últimos tendem a perder sua intensidade e nivelar-se quando as escalas de tempo são modificadas (LOKSHINA *et al.*, 2020).

Essa descoberta de Leland *et al.* (1994) representou um marco importante na compreensão da natureza do tráfego e levou a uma mudança de paradigma na modelagem de processos de tráfego e redes.

De uma perspectiva abrangente, é possível conceituar o tráfego de uma rede como um processo fractal, uma estrutura na qual fragmentos diminutos, quando infinitamente ampliados, assemelham-se à configuração original da rede, conforme destacado por Lysenko *et al.* (2020). Esse processo fractal, por sua vez, é caracterizado por ser aleatório, exibindo propriedades estatísticas de autossimilaridade. Essa traduz a invariância de escala, como discutido por Barsukov *et al.* (2019). Em termos mais simples, a complexidade intrínseca do tráfego de rede revela padrões repetitivos em diferentes escalas.

A invariância de escala pode ser observada nos gráficos da Figura 4, que representam alguns minutos de monitoramento contínuo do tráfego de uma rede residencial em Crateús, durante junho de 2021. Cada gráfico exibe 245 pontos de dados em escalas de tempo diferentes (1ms, 10ms, 100ms e 1s). É notável que a forma do tráfego se mantém semelhante em todas essas quatro escalas de tempo distintas.

Figura 4 – Captura de junho de 2021 do tráfego de uma rede residencial de Crateús em 4 escalas de tempo diferentes: (a) 1ms - (b) 10ms - (c) 100ms - (d) 1s.



Fonte: Elaborada pelo autor.

Dado que a autossimilaridade é um processo estocástico, o grau de autossimilaridade pode ser quantificado através do uso do Parâmetro de Hurst, que é uma ferramenta fundamental para analisar séries temporais nas quais os dados de tráfego de rede foram coletados (LYSENKO *et al.*, 2020). Tal parâmetro, representado por H, constitui a característica numérica mais fundamental para descrever a autossimilaridade estocástica e a LRD (JEONG *et al.*, 2017). Sua definição é a seguinte:

a) Quando o valor do H está no intervalo de 0 a $0,5$, isso indica que os eventos são aleatórios e não há uma LRD entre eles, caracterizando a presença de uma SRD;
b) Por outro lado, quando o valor do H está no intervalo de $0,5$ a 1, isso significa que o intervalo de tempo observado constitui uma série temporal contínua com LRD.

Portanto, para que o tráfego seja classificado como autossimilar, o cálculo do H deve estar dentro da faixa de $0,5$ a 1, sendo que valores mais próximos de 1 indicam uma autossimilaridade mais pronunciada (LYSENKO *et al.*, 2020). A natureza autossimilar do tráfego já foi identificada em diversos sistemas, incluindo o tráfego de longa distância (*Wide Area*) (PAXSON; FLOYD, 1995), a *World Wide Web* (CROVELLA; BESTAVROS, 1997) e redes *Asynchronous Transfer Mode* (ATM) (TSYBAKOV; GEORGANAS, 1998).

Uma das consequências significativas e relevantes do cálculo de H no tráfego de rede é a capacidade de detectar ataques e outras anomalias por meio da análise fractal desse tráfego. A análise fractal permite a identificação de características estruturais anormais que se diferenciam do comportamento típico do tráfego normal (BARSUKOV *et al.*, 2019). Portanto, ao monitorar as alterações no H, é possível identificar e determinar a ocorrência de ataques à rede. Isso permite que as organizações estejam preparadas para reagir a mudanças repentinas no valor do expoente, que indicam comportamento incomum na rede.

O trabalho de Li (2004) destaca que quando ocorre um ataque DDoS, o valor do H no tráfego de rede sofre mudanças. De fato, vários estudos, como os de Deka e Bhattacharyya (2016) e Yu *et al.* (2016), identificaram que durante o aumento da intensidade do tráfego devido a ataques de DDoS, o H pode assumir valores fora do habitual. Essa alteração na natureza do tráfego ocorre porque os ataques de DDoS e sua detecção são fenômenos de curto alcance (KAUR *et al.*, 2020).

Com base nisso, a detecção eficaz de ataques de DDoS pode ser realizada monitorando a variação dos valores do H. Isso pode ser feito identificando desvios repentinos em

relação ao comportamento normal do tráfego, como relatado em Li *et al.* (2020b). É importante destacar que a detecção de anomalias usando o cálculo do H no tráfego de rede é um método leve, uma vez que não envolve a inspeção do conteúdo dos pacotes.

3 TRABALHOS RELACIONADOS

Neste capítulo, apresentaremos os trabalhos mais relevantes na literatura em relação aos objetivos da pesquisa atual. Esta seção está dividida em duas partes: a primeira aborda os estudos que utilizaram o Parâmetro de Hurst para detecção de ataques de DDoS, enquanto a segunda parte se concentra nas pesquisas que empregaram a SDN para proteger as redes IoT contra ataques de DDoS. O Quadro 1 fornece uma visão geral dos trabalhos comparados neste contexto, destacando os conceitos abordados (Hurst, IoT e SDN), se realizam detecção dos ataques e se os experimentos foram validados em redes com ou sem fio.

Quadro 1 – Visão geral dos trabalhos.

Trabalho	**Hurst (H)**	**IoT**	**SDN**	**Detecção**	**Com Fio | Sem Fio**
Deka e Bhattacharyya (2016)	Sim	Não	Não	Sim	Com fio
Yu *et al.* (2016)	Sim	Não	Não	Sim	Com fio
Kotenko *et al.* (2020)	Sim	Sim	Não	Sim	Ambos
Li *et al.* (2020b)	Sim	Não	Sim	Sim	Com fio
Bull *et al.* (2016)	Não	Sim	Sim	Sim	Com fio
Yin *et al.* (2018)	Não	Sim	Sim	Sim	Sem fio
Yan *et al.* (2018)	Não	Sim	Sim	Sim	Com fio
Ravi e Shalinie (2020)	Não	Sim	Sim	Sim	Ambos
Chen *et al.* (2020b)	Não	Sim	Sim	Sim	Ambos
Cheng *et al.* (2020)	Não	Sim	Sim	Sim	Ambos
Silveira *et al.* (2020)	Não	Sim	Sim	Sim	Sem fio
Wang *et al.* (2021)	Não	Sim	Sim	Sim	Ambos
Hafeez *et al.* (2020)	Não	Sim	Sim	Sim	Sem fio
Galeano-Brajones *et al.* (2020)	Não	Sim	Sim	Sim	Com fio
Este trabalho	Sim	Sim	Sim	Sim	Sem fio

Fonte: Elaborado pelo autor.

3.1 Parâmetro de Hurst na detecção de DDoS

Conforme mencionado, a primeira pesquisa a ser discutida é o trabalho de Deka e Bhattacharyya (2016), que desenvolveu um método para a detecção de ataques de DDoS baseado na propriedade de autossimilaridade do tráfego de rede. O objetivo principal deste método era identificar ataques de DDoS, independentemente de sua intensidade. O framework criado realizava o cálculo do H para detectar um ataque DDoS quando o resultado estava fora do intervalo de $0,5$ a 1. Para validar a eficácia do método, foram conduzidos estudos experimentais utilizando *datasets*, e os resultados demonstraram que o cálculo de H é uma ferramenta útil para distinguir o tráfego de ataques de DDoS do tráfego legítimo. Esta abordagem representa um avanço na detecção de ataques de DDoS, uma vez que utiliza a autossimilaridade do tráfego

de rede como uma métrica eficaz para identificar esses ataques, independentemente de sua magnitude.

Continuando na mesma direção, o trabalho de Yu *et al.* (2016) desenvolveu um algoritmo que se baseia na estimativa da autossimilaridade para detectar eventos anômalos em sistemas e redes. Similar ao estudo anterior, esse trabalho também utilizou *datasets* em seus experimentos. No entanto, além da análise de *datasets*, o método foi implementado em um Sistema de Detecção de Intrusão (IDS). Os resultados dos experimentos demonstraram que essa abordagem é eficaz e intuitiva na detecção de uma variedade de ataques de maneira eficaz. Essa pesquisa destaca a utilidade da autossimilaridade na detecção de eventos anômalos em sistemas e redes, fornecendo uma abordagem promissora para melhorar a segurança cibernética por meio da identificação de atividades suspeitas.

No trabalho conduzido por Kotenko *et al.* (2020), uma abordagem foi proposta para a detecção de ciberataques em uma rede *Smart Grid*, com foco na identificação de anomalias no tráfego por meio da avaliação de sua autossimilaridade. Considerando que a *Smart Grid* é uma rede IoT, a pesquisa realizou um estudo para confirmar a presença da autossimilaridade nesse contexto. Ao confirmar a presença da autossimilaridade, a pesquisa identificou que a ocorrência de anomalias na rede, causadas por ataques como DDoS, resulta em uma mudança significativa no parâmetro de Hurst.

Para validar a abordagem proposta, os pesquisadores desenvolveram um protótipo que implementa o método e criaram um *dataset* com tráfego real contendo ataques cibernéticos. Em seguida, conduziram experimentos que comprovaram a eficácia do método proposto na detecção de ciberataques na rede *Smart Grid*. Essa pesquisa destaca a aplicabilidade do cálculo do H na detecção de anomalias e ataques em redes IoT, contribuindo para a segurança cibernética desses ambientes.

O estudo conduzido por Li *et al.* (2020b) se concentrou na segurança das redes SDN e apresentou o SA-Detector, uma ferramenta projetada para detectar ataques de saturação, frequentemente implementados por diversos tipos de ataques de DDoS em ambientes SDN. O método subjacente ao SA-Detector se baseia nas discrepâncias na autossimilaridade entre os fluxos de tráfego e na proporção do número de pacotes OpenFlow. O SA-Detector foi implementado em duas versões: uma *online* e outra *offline*. Na versão *online*, a ferramenta é instalada diretamente no controlador, permitindo que trabalhe em conjunto com a SDN para proteger a rede em tempo real. Essa abordagem demonstra um compromisso com a detecção

de ataques de DDoS em redes SDN, permitindo que a proteção seja adaptada às necessidades específicas da rede e da organização.

Com base na análise dos trabalhos apresentados, fica evidente que o cálculo do Hurst é uma ferramenta eficaz na detecção de ataques de DDoS, inclusive em redes IoT. Esses estudos demonstram que o parâmetro de Hurst no tráfego de rede pode ser utilizada como uma métrica confiável para identificar anomalias e ataques, oferecendo uma abordagem viável para aplicação na IoT.

A capacidade de detectar ataques de DDoS na IoT por meio do H do tráfego é promissora, uma vez que a IoT está se tornando cada vez mais integrada em nosso cotidiano, e a segurança dessas redes é uma preocupação. Ao aplicar métodos baseados em H, é possível melhorar a capacidade de detecção precoce de ataques, permitindo respostas mais eficazes para proteger a integridade das redes IoT e dos dispositivos conectados.

3.2 SDN na proteção IoT contra DDoS

Nesta seção, discutiremos trabalhos relevantes que empregaram a SDN para proteger a IoT contra ataques e DDoS, abordando tanto a detecção quanto a mitigação dessas ameaças.

Começando com o estudo de Bull *et al.* (2016), foi apresentada uma abordagem de segurança baseada em fluxo adaptativo para dispositivos IoT, utilizando um *gateway* SDN. Esse mecanismo adaptativo realiza análises dinâmicas dos padrões de tráfego dos dispositivos IoT, visando identificar comportamentos maliciosos ou tentativas de exploração externa. Para atingir esse propósito, faz uso do *gateway* SDN para detecção de atividades anômalas e a subsequente implementação de medidas apropriadas em relação ao tráfego suspeito. Os resultados desse estudo demonstraram que essa abordagem, que combina SDN e IoT, foi capaz de efetivamente identificar e bloquear ataques de inundação.

Explorando mais a integração entre SDN e IoT, o estudo realizado por Yin *et al.* (2018) apresenta um *framework* abrangente para a IoT Definida por *Software* (SD-IoT). Nesse contexto, o estudo propõe um algoritmo destinado a identificar a ocorrência de ataques de DDoS, rastrear a origem real do ataque e aplicar bloqueios na sua fonte. Os resultados obtidos demonstram que o algoritmo incorporado ao *framework* é capaz de identificar prontamente o dispositivo IoT a partir do qual um ataque DDoS é lançado em um curto espaço de tempo.

Com foco na segurança da Internet Industrial das Coisas (IIoT), Yan *et al.* (2018) também propôs um *framework* abrangente para mitigação de ataques de DDoS, organizado em

múltiplos níveis, incluindo *Edge*, *Fog* e *Cloud Computing*. No nível de *Edge Computing*, são utilizados *gateways* IIoT baseados em SDN para gerenciar e proteger os nós da rede. No nível de *Fog Computing*, é implementada uma unidade de controle de gerenciamento IIoT que faz uso de um *cluster* de controladores e aplicativos SDN para detecção e neutralização de ataques de DDoS. Por fim, no nível de *Cloud Computing*, são aplicadas técnicas de *big data* e *smart computing* para analisar o tráfego de rede. Os resultados obtidos demonstram que a combinação desses três níveis de computação, aliada à capacidade de programação de rede oferecida pela SDN, mostra-se promissora na resolução do problema de ataques de DDoS.

Outra abordagem para fortalecer a segurança na IoT foi desenvolvida por Ravi e Shalinie (2020), que criou o *Learning-driven Detection Mitigation Mechanism* (LEDEM). Esse mecanismo se destaca por sua capacidade de detecção de ataques de DDoS por meio de um algoritmo de aprendizado de máquina supervisionado. No contexto da arquitetura, uma SDN descentralizada de duas camadas foi adotada, com um controlador universal conectado a vários controladores locais. É nesses controladores locais que a detecção de ataques ocorre, permitindo uma abordagem mais eficaz para a defesa contra ameaças na IoT.

De forma bastante semelhante ao LEDEM de Ravi e Shalinie (2020), o estudo conduzido por Chen *et al.* (2020b) apresentou um sistema de detecção de ataques DDoS em múltiplas camadas, também fundamentado em aprendizado de máquina, para prevenir ataques no *gateway* IoT. A principal distinção entre essas abordagens reside na arquitetura. No trabalho de Chen *et al.* (2020b), não é adotado um controlador universal na nuvem, diferenciando-se assim do LEDEM. No entanto, este estudo foi além ao realizar uma implementação prática, empregando diferentes protocolos de comunicação para validar sua proposta.

Seguindo uma abordagem semelhante às anteriores, Silveira *et al.* (2020) apresentou o sistema *Smart Detection-IoT*, uma solução que faz uso de aprendizado de máquina para categorizar o tráfego de rede na IoT e identificar ataques de negação de serviço. Notavelmente, essa detecção é realizada exclusivamente analisando os cabeçalhos de amostras de tráfego de rede, o que não compromete a privacidade dos dados transmitidos. O objetivo central é identificar a ameaça o mais próximo possível de sua origem, permitindo ação imediata. A abordagem de detecção proposta opera por meio de um sensor instalado no ponto de acesso da rede IoT, o qual classifica amostras de tráfego de forma aleatória, possibilitando identificar e responder a ameaças de maneira eficaz.

Em outra perspectiva, com foco na aplicação de aprendizado de máquina na detecção,

Cheng *et al.* (2020) introduziu um mecanismo projetado para distinguir o tráfego normal do tráfego de baixa taxa de ataques de DDoS. Esse mecanismo atua diretamente no controlador SDN e nos *switches* em um ambiente de rede IoT que está integrado à SDN. É importante observar que essa abordagem se concentra especificamente em ataques de baixa intensidade e pressupõe que a rede IoT já esteja integrada com a SDN, uma vez que seu método de detecção faz uso de recursos dos pacotes OpenFlow para classificar o tráfego.

Seguindo a abordagem de trabalhar com IoT já integrada à SDN, Wang *et al.* (2021) conduziu testes e análises do algoritmo *SDN Secure Control and Data Plane* (SECOD), desenvolvido pelos próprios autores. O objetivo principal desse algoritmo é proteger redes IoT baseadas em SDN contra ataques de DDoS, fazendo uso de limites de tempo nas mensagens OpenFlow para detecção de ameaças. O algoritmo SECOD proposto é executado na camada de aplicação da arquitetura SDN. As políticas predefinidas nele são convertidas em entradas de fluxo por meio do controlador e, posteriormente, são inseridas nos *switches* OpenFlow. Além disso, os autores demonstraram a adaptabilidade do SECOD ao modificá-lo para obter um desempenho aprimorado em uma variedade de cenários. Essa flexibilidade é fundamental para lidar eficazmente com diferentes contextos de segurança na IoT integrada à SDN.

Em seu trabalho, Hafeez *et al.* (2020) apresenta o IOT-KEEPER, um sistema desenvolvido para detectar vários tipos de ataques à rede, incluindo DDoS, e aplicar as medidas de segurança necessárias para impedir que os dispositivos IoT executem esses ataques. O principal objetivo desse sistema é proteger a comunicação da IoT em redes de borda, para o qual utiliza um algoritmo de *clustering* implementado no controlador SDN. É importante notar que o controlador desempenha um papel fundamental no IOT-KEEPER, abrigando todos os módulos do sistema, incluindo monitoramento, detecção e imposição de medidas de segurança. Esse sistema demonstra uma abordagem completa e integrada para garantir a segurança das redes IoT.

Com o objetivo de detectar ataques de DDoS na IoT, Galeano-Brajones *et al.* (2020) concebeu uma solução de segurança que faz uso da entropia em conjunto com a tecnologia SDN. Nesta abordagem, a entropia é empregada como um método de detecção, aproveitando seu caráter estatístico simplificado. A SDN, por sua vez, é encarregada de combater os ataques de DDoS, configurando regras distintas nos *switches* para descartar pacotes suspeitos. Isso é alcançado por meio de um algoritmo implementado no controlador SDN. É relevante destacar esse trabalho, pois ele apresenta uma solução similar à que está sendo considerada em nosso projeto, embora com a diferença fundamental de utilizar a entropia como abordagem estatística

em vez do cálculo do H.

É evidente que a pesquisa na área de segurança para IoT, especialmente no contexto de ataques de DDoS, tem adotado diversas abordagens e técnicas. O aprendizado de máquina e métodos estatísticos têm se mostrado ferramentas valiosas para a detecção de ameaças, permitindo a identificação de padrões anômalos no tráfego de rede. O controlador SDN desempenha um papel crucial nesse cenário, pois oferece um ponto centralizado para implementar estratégias de detecção. Sua capacidade de gerenciar a rede e tomar decisões dinâmicas com base em políticas e algoritmos o torna um local estratégico para a proteção da IoT contra ameaças.

A abordagem de explorar o parâmetro de Hurst na detecção de ataques deDDoS na rede IoT é interessante e inovadora. A utilização desse método estatístico pode trazer novas perspectivas para a detecção de ameaças, uma vez que está relacionado à análise de séries temporais e pode revelar comportamentos anômalos no tráfego de rede. A combinação dessa abordagem com a tecnologia SDN, tem o potencial de criar um sistema automatizado de detecção de DDoS ainda mais robusto e adaptável. A SDN oferece a flexibilidade necessária para reagir rapidamente a ameaças detectadas.

Essa proposta de sistema automatizado de detecção de DDoS, que integra o cálculo do Hurst e a SDN, realmente parece ser promissora. Podendo representar um avanço na proteção de redes IoT contra ataques de DDoS, uma vez que combina um método estatístico com o poder de controle e adaptação da SDN.

4 METODOLOGIA

No capítulo de Metodologia, delineamos sobre as escolhas e estratégias adotadas na condução da pesquisa, oferecendo uma visão transparente do processo de desenvolvimento do presente projeto.

Inicialmente, realizou-se uma pesquisa bibliográfica utilizando principalmente as ferramentas *Google* Acadêmico, *IEEE Xplore* e o Portal de Periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), em busca de trabalhos que abordassem os termos *self-similarity* e *IoT security* no período de 2010 a 2022. A busca por esses termos está relacionada ao interesse do pesquisador pelos temas de autossimilaridade no tráfego de rede, o qual mede-se matematicamente usando o Parâmetro de Hurst, e segurança em IoT. Durante as primeiras leituras, percebeu-se que a SDN emergia como uma solução a ser explorada na segurança da IoT. Assim, o termo foi incluído na busca.

Após ler mais artigos, foi escolhido o tema de pesquisa ao constatar que os ataques de DDoS representam um problema crescente nas redes IoT e que o cálculo de Hurst do tráfego pode ser utilizada para detectar esse tipo de ataque. Além disso, identificamos que a SDN seria útil na integração com a IoT, desempenhando um papel na coleta de dados para a detecção de ataques de DDoS.

Para isso, foram lidos mais trabalhos a fim de definir e aprimorar a proposta do projeto(detalhado no Capítulo 5). Assim, idealizamos um sistema voltado para detecção de ataques de DDoS em redes IoT.

Com a proposta definida, iniciamos a configuração dos experimentos. Optamos pelo *Mininet-WiFi*, uma ferramenta que emula cenários SDN sem fio, possibilitando experimentos de alta fidelidade que replicam ambientes de rede semelhantes aos reais (FONTES *et al.*, 2015). Escolhemos o *Ryu* como controlador SDN, caracterizado por seu código aberto, baseado em componentes, e uma API bem definida que simplifica a criação de novos aplicativos de gerenciamento e controle de rede (RYU, 2022). O *Ryu*, desenvolvido em *Python*, assim como o *Mininet-WiFi*, é amplamente adotado pela comunidade de pesquisa (GALEANO-BRAJONES *et al.*, 2020). Por fim, detalhamos o cenário, as métricas, os fatores, os níveis e a execução dos experimentos (ver Capítulo 6 para mais detalhes).

Com os experimentos devidamente definidos, procedemos à construção dos cenários de teste, os quais consistem em redes de sensores utilizando o protocolo *Message Queuing Telemetry Transport* (MQTT). Esta etapa envolveu a implementação prática das configurações

delineadas, utilizando as ferramentas selecionadas, o *Mininet-WiFi* e o *Ryu*), e *scripts* automatizados, que permitem a criação e comunicação dos nós na rede emulada. A construção dos cenários visou criar ambientes representativos, permitindo a execução de experimentos que reproduzissem as condições encontradas em redes reais.

Após a construção dos cenários experimentais, deu-se início ao desenvolvimento do sistema proposto. Este estágio marcou a transição da fase preparatória para a implementação prática da solução, permitindo a validação e análise dos resultados obtidos em ambiente controlado. O sistema, concebido para atender aos objetivos delineados, foi elaborado de forma modular, contento um módulo de coleta de dados do tráfego e outro para detectar ataques a partir do cálculo de Hurst (Ver detalhadamente em Capítulo 5).

Com o sistema proposto devidamente desenvolvido, avançamos para a fase dos testes funcionais. Essa etapa visava validar o sistema em condições práticas, assegurando que atenda às especificações e requisitos previamente estabelecidos. Os testes funcionais foram fundamentais para identificar e corrigir eventuais falhas, garantindo a melhoria do sistema antes da sua implementação completa.

Após a conclusão dos testes funcionais, iniciamos a fase de refinamento do sistema. Nessa etapa, concentramo-nos em aprimorar aspectos identificados durante os testes, visando otimizar o desempenho, a eficiência e a usabilidade do sistema. As observações e *feedbacks* obtidos durante os testes funcionais foram importantes para direcionar as melhorias necessárias, garantindo que o sistema atenda de maneira mais eficaz às necessidades.

Posteriormente ao refinamento do sistema, iniciamos a fase de avaliação de desempenho. Nessa etapa, buscamos mensurar e analisar o comportamento do sistema em condições diversas. Utilizando métricas específicas e cenários representativos, tal avaliação visava fornecer uma compreensão clara e objetiva do desempenho do sistema.

Logo após a avaliação de desempenho do sistema, adentramos a fase de análise dos resultados. Neste estágio, examinamos os dados obtidos durante os experimentos. Por fim, relatamos as conclusões obtidas no presente trabalho, além dos aprendizados e caminhos a serem seguidos no futuro.

5 PROPOSTA

Levando em conta a detecção de ataques pelo cálculo de Hurst e a programabilidade advinda da integração entre SDN e IoT, apresentamos o sistema denominado *DDoS Detection System based on Hurst Parameter* (DDSHP).

Ao contemplar a detecção de ataques através da análise do cálculo de Hurst e a capacidade programável decorrente da integração entre SDN e IoT, introduzimos o sistema conhecido como *DDoS Detection System based on Hurst Parameter* (DDSHP). Este sistema representa uma abordagem para identificar ataques de DDoS, capitalizando a análise do parâmetro de Hurst para discernir padrões de tráfego anômalos. A sinergia entre SDN e IoT proporciona uma flexibilidade adicional, permitindo coletas dinâmicas do tráfego.

De modo geral, buscamos a aplicabilidade do em diversos cenários da IoT, independentemente dos protocolos (WiFi, 5G, Zigbee, etc.) e com implementação flexível. Além disso, almejamos que o sistema utilize o paradigma da SDN para coletar os dados da rede IoT durante o seu funcionamento. Isso implica aproveitar as vantagens inerentes da SDN, como análise de tráfego baseada em software, controle lógico centralizado, inserção/exclusão dinâmica de fluxos em *switches* e visão global da rede.

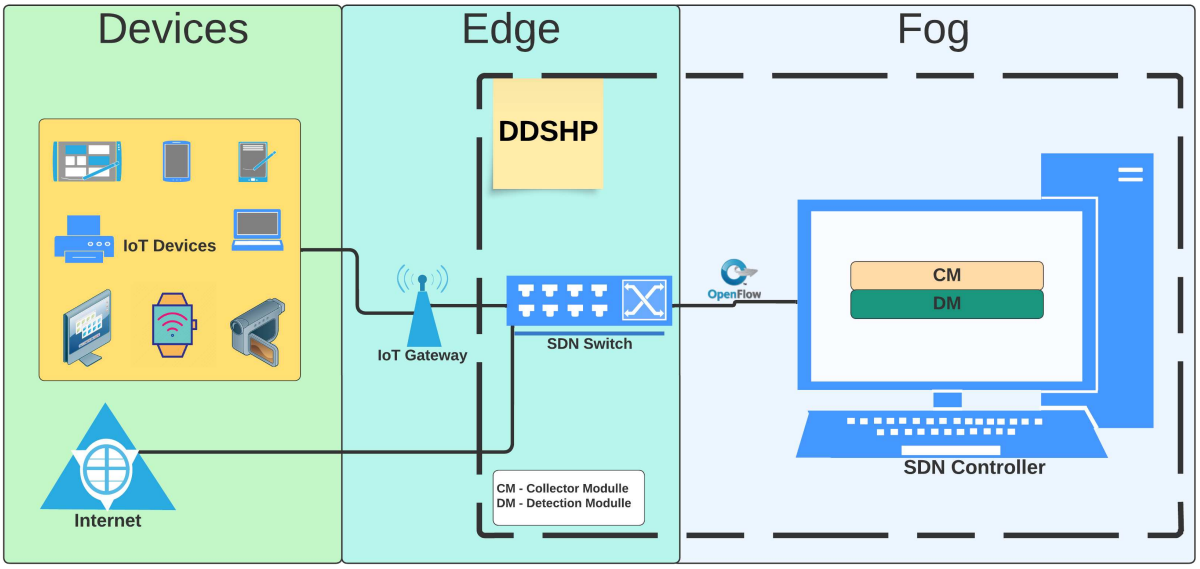
O DDSHP calculará Hurst para detectar ataques. O sistema realizará uma análise estatística comparativa entre o comportamento normal dos fluxos e os desvios temporais, utilizando como referência a *baseline*, que, neste caso, é o Parâmetro de Hurst. Além disso, como evidenciado, o sistema é uma solução centrada no tráfego, concentrando-se nos fluxos para identificar atividades maliciosas.

A seguir, apresentamos uma análise detalhada da infraestrutura do DDSHP, destacando seus componentes e descrevendo suas funções específicas. O objetivo deste exame é proporcionar uma compreensão clara do arcabouço que sustenta o DDSHP, evidenciando a estrutura fundamental e os propósitos distintos de cada componente. Além disso, discutiremos os requisitos para estabelecer a *baseline* do Hurst normal da rede no DDSHP e sua, posterior, definição no sistema. Por fim, abordaremos o funcionamento do sistema após a configuração da *baseline*.

5.1 Infraestrutura

No âmbito da detecção de ataques de DDoS em redes IoT, elaboramos o DDSHP, composto por dois módulos fundamentais: o *Collector Module* (CM) e o *Detection Module* (DM). O CM é responsável por coletar e analisar dados de tráfego, enquanto o DM utiliza essas informações para identificar padrões suspeitos e potenciais ataques. A Figura 5 oferece uma representação visual da infraestrutura do sistema, com ênfase na integração com a SDN. Este arranjo estratégico visa aprimorar a eficácia da detecção e resposta a DDoS, proporcionando uma abordagem abrangente e eficiente no contexto específico da Internet das Coisas.

Figura 5 – Infraestrutura do DDSHP.



Fonte: Elaborada pelo autor.

O DDSHP utiliza o *switch* para receber e encaminhar o tráfego dos dispositivos, tendo o IoT *gateway* como o intermediário deles. O sistema aplica o OpenFlow para facilitar a comunicação entre o *switch* e o controlador, que possui uma visão geral da rede e armazena os módulos do DDSHP. Cada módulo desempenha uma função específica, que será descrita detalhadamente a seguir:

a) ***Collector Module* (CM)**: Monitora continuamente o tráfego e coleta estatísticas de fluxos geradas pelos dispositivos de encaminhamento (*switches*) usando o OpenFlow. Essas estatísticas referem-se à contagem de pacotes que chegaram dentro de um período específico e são essenciais para o cálculo do expoente de Hurst realizado no DM;

b) ***Detection Module* (DM)**: Calcula o Hurst e identifica a ocorrência de um ataque

de DDoS. Recebe as contagens de pacotes do CM e calcula o parâmetro de Hurst por meio da Análise R/S, um método amplamente utilizado e simples (PATIL *et al.*, 2011). Após obter o valor de Hurst, verifica se está dentro do padrão normal de tráfego. Caso contrário, considera a ocorrência de um ataque.

Outro ponto a ser destacado é a localização das partes integrantes do sistema. No atual contexto, em que a computação em névoa desempenha um papel destacado como elo entre IoT, nuvem e computação de borda (DONNO *et al.*, 2019), é necessário identificar a localização específica de cada componente do sistema.

Seguindo o conceito de computação em borda de aproximar o processamento de dados dos dispositivos IoT, além de que a computação em névoa atua como uma camada intermediária entre a IoT e a computação em nuvem (DONNO *et al.*, 2019), conclui-se o seguinte:

a) Os *switches* SDN ficam na borda da rede, ou seja, na *edge*. Afinal estão próximos dos dispositivos IoT e dos IoT *gateways*, além de exigirem um certo poder de processamento;
b) O controlador permanece na *fog*, por exigir maior poder de processamento em comparação com os *switches*. No entanto, não pode afastar-se dos dispositivos finais devido à latência.

Destacamos que nosso sistema pode utilizar mais de um *switch*, dependendo da rede. Além disso, é viável ter múltiplos controladores, cada um contendo ambos os módulos ou apenas um módulo específico. Isso assegura a flexibilidade na instalação do DDSHP. A infraestrutura apresentada revela o potencial do DDSHP ao dividir as funções em módulos, possibilitando a implementação do sistema de diversas maneiras, adaptando-se às exigências dos ambientes IoT.

5.2 Configuração do Sistema

A configuração do DDSHP é uma etapa importante para o seu correto funcionamento, requerendo uma análise do tráfego de rede. O ponto aqui é sobre os intervalos a serem considerados para o processo de detecção. É imperativo estabelecer critérios explícitos para a detecção, determinando valores aceitáveis para os limiares do Parâmetro de Hurst do tráfego normal da rede. Consideramos, então, a utilização do intervalo de confiança, o qual para ser criado necessita de um estudo preliminar da rede.

Para estabelecer o intervalo de confiança do parâmetro de Hurst, devem ser coletados

no mínimo 30 valores desse a partir do tráfego normal da rede. Essa coleta pode ser realizada por meio do sistema proposto, DDSHP, que calcula e retorna os valores de Hurst, sendo capaz de realizar esse cálculo a cada minuto. Dessa forma, uma hora de execução do sistema já proporciona uma quantidade substancial de dados, permitindo a geração confiável do intervalo de confiança e, consequente, configuração do sistema proposto, mais precisamente do módulo de detecção (DM).

Adicionalmente, o método proposto destaca a importância do estudo preliminar em cada rede para ajustar o módulo de detecção. Essa adaptação personalizada dos parâmetros do sistema à dinâmica específica de cada ambiente de rede contribuirá para a flexibilidade e eficiência do DDSHP. Dessa forma, garantimos que a configuração do sistema não apenas atenda aos critérios gerais, mas também se ajuste de forma otimizada aos desafios e características singulares de diferentes contextos.

Ao estabelecer os limiares de detecção no contexto do DDSHP, é importante reconhecer que o intervalo de confiança não deve ser o único critério considerado. Além desse parâmetro, outros aspectos, como a média do valor, desempenham um papel que pode ser relevante. A análise conjunta desses fatores proporciona uma abordagem mais abrangente e robusta na definição dos limiares de detecção, promovendo uma configuração mais refinada do sistema.

Em síntese, esta seção não só delineia os princípios fundamentais da configuração do DDSHP, ou seja, da definição da *baseline*, mas também implica no estudo detalhado da rede antes de configurar o sistema. Na Tabela 3 podemos observar resumidamente o que deve ser feito para configurar o DDSHP antes de sua utilização.

Tabela 3 – Etapas de configuração do DDSHP.

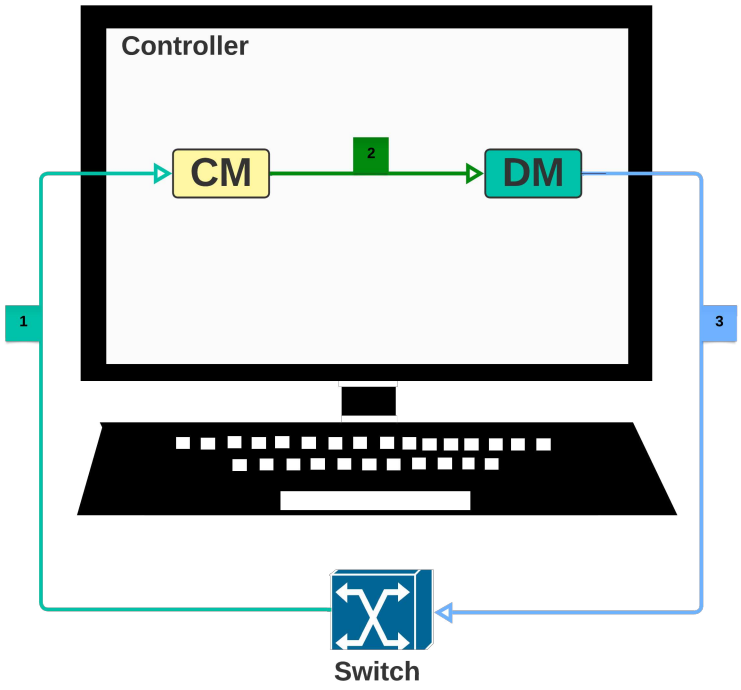
Etapas	**Descrição**
Estudo do tráfego de rede	Capturar pelo menos 30 valores de Hurst do tráfego de rede para estabelecer um intervalo de confiança.
Definição da *baseline*	Analisar os resultados do intervalo de confiança e outras estatísticas para estabelecer uma *baseline* do tráfego normal, incluindo os limites superior e inferior de Hurst.
Inserção da *baseline* no DDSHP	Inserir a *baseline* de Hurst do tráfego normal da rede definida no DM, para considerar qualquer valor fora desse intervalo como anômalo.

Fonte: Elaborado pelo autor.

5.3 Funcionamento

Como mencionado na Seção 5.1, cada módulo desempenha uma função específica, colaborando para monitorar o tráfego de rede e identificar atividades maliciosas. Na Figura 6, é possível visualizar o fluxo de trabalho do DDSHP, dividido em três etapas de comunicação. Observa-se que o desempenho eficaz do DDSHP está diretamente ligado à comunicação entre seus módulos, seja em um único controlador ou em controladores distintos, e à comunicação *switch*-controlador, seguindo o padrão do SDN. Isso, além da configuração adequada do módulo de detecção com o padrão de tráfego definido como normal.

Figura 6 – Fluxo de trabalho do DDSHP.



Fonte: Elaborada pelo autor.

Na primeira etapa, ocorre a comunicação *switch*-CM. O *switch*, por meio do OpenFlow, gera estatísticas de fluxo e as envia para o CM. Este, por sua vez, monitora o tráfego ao coletar essas estatísticas, que representam a contagem de pacotes recebidos a cada 0,5 segundos, e as envia para o DM a cada minuto. Esses dados são essenciais para o cálculo do expoente de Hurst a ser realizado.

A segunda etapa envolve o envio da contagem de pacotes realizada pelo CM para o DM. Este módulo, então, calcula o parâmetro de Hurst por meio da Análise R/S, um dos métodos mais usados e simples para esse cálculo (PATIL *et al.*, 2011). Com o valor de Hurst obtido, verifica-se se está dentro dos limites definidos como tráfego normal. Por exemplo, se o

estudo preliminar da rede (discutido na Seção 5.2) estabelecer que o Hurst do tráfego normal está no intervalo de 0,4 a 0,7, então um valor dentro desse intervalo é considerado normal. Caso contrário, se estiver fora do intervalo, considera-se a ocorrência de um ataque.

Com a confirmação de um ataque de DDoS, a terceira etapa se inicia. O DM ativa o *switch*, alertando-o sobre o ataque e, por consequência, notificando o administrador de rede. Este tem a responsabilidade de identificar os endereços dos dispositivos atacantes e pode instalar regras de fluxo no *switch* para bloquear o tráfego desses dispositivos, eliminando assim sua influência na rede.

Portanto, o DDSHP opera ao permitir que o *switch* forneça informações de fluxo ao controlador SDN. Este, por sua vez, utiliza tais dados em conjunto com os módulos para distinguir o tráfego como malicioso ou benigno. Posteriormente, o controlador emprega o OpenFlow para comunicar ao *switch* e ao administrador, instruindo a prevenção de fluxos associados a possíveis ataques.

Por isso, ressaltamos que qualquer dispositivo crucial para a disponibilidade da rede, como um servidor, deve estar conectado ao *switch* SDN. Este dispositivo será o alvo potencial de um ataque, e, como o *switch* que emitirá o aviso de ataque, todo o tráfego direcionado à entidade principal deve passar por ele.

Conforme apresentado, o OpenFlow é o ideal para o nosso sistema. Um *switch* habilitado para OpenFlow baseia-se em tabelas de fluxo, cada uma com regras, ações para pacotes e contadores para estatísticas de pacotes (KREUTZ *et al.*, 2015). Esse modelo simplificado, derivado do OpenFlow e amplamente difundido na SDN, atende às necessidades do DDSHP, principalmente na coleta de estatísticas.

Em síntese, destacamos que as duas fases iniciais do funcionamento ocorrem de forma contínua ao longo da atividade do sistema, desempenhando papéis fundamentais na sua operação regular. Por outro lado, a última etapa entra em cena somente quando há a detecção de um potencial ataque na rede. Nesse contexto, a compreensão clara das diferentes etapas contribui para uma visão abrangente do funcionamento do sistema em ambientes diversos.

6 DESENVOLVIMENTO

No decorrer deste capítulo, abordamos o desenvolvimento da proposta formulada, explorando os experimentos realizados. O objetivo principal é a análise aprofundada de aspectos pertinentes ao DDSHP, levando em consideração não apenas os objetivos traçados, mas também a eficácia e a eficiência com que as funções são desempenhadas. Ao examinar cada etapa do processo de desenvolvimento e os resultados obtidos nos experimentos, busca-se fornecer uma compreensão abrangente e embasada sobre a contribuição do DDSHP no contexto de segurança, ampliando assim o conhecimento no campo em questão.

6.1 Escopo

No capítulo anterior, abordamos a definição da infraestrutura, destacando a essência do DDSHP, cujo principal propósito é a detecção de ataques DDoS em ambientes IoT. Este sistema adota uma abordagem estratégica, valendo-se da tecnologia SDN para a coleta eficiente de estatísticas de tráfego. Além disso, faz uso do cálculo de Hurst como uma ferramenta para identificar possíveis anomalias na rede. Neste capítulo, detalharemos de forma precisa e concisa a implementação prática desses métodos, proporcionando uma compreensão aprofundada do processo de detecção de ameaças DDoS no contexto específico dos dispositivos IoT.

Como destacado no capítulo anterior, que detalha a infraestrutura, o DDSHP visa primariamente identificar potenciais ataques de DDoS em ambientes IoT. Essa finalidade é alcançada por meio da aplicação da SDN para a coleta de estatísticas de tráfego. Além disso, o método incorpora o cálculo de Hurst, para a detecção de anomalias, proporcionando a identificação de padrões incomuns que possam indicar atividades maliciosas na rede IoT.

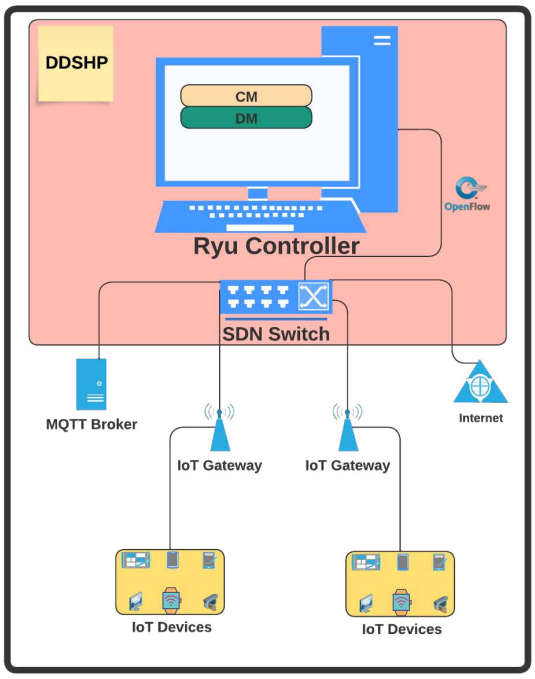
Dessa forma, elaboramos três cenários que viabilizam a emulação de redes IoT com distintos tamanhos, proporcionando uma análise abrangente do desempenho do DDSHP na detecção de ataques em ambientes diversos. A criação desses cenários representa permite avaliar a eficácia do sistema diante de cenários heterogêneos. A seguir, apresentamos cada um desses cenários:

a) **Cenário 1 (*Smart Home*)**: 20 dispositivos IoT, 1 servidor IoT, 1 controlador SDN, 2 IoT *gateways* e 1 *switch* SDN;
b) **Cenário 2 (*Smart Market*)**: 50 dispositivos IoT, 1 servidor IoT, 1 controlador SDN, 5 IoT *gateways* e 1 *switch* SDN;

c) **Cenário 3 (*Smart Hospital*)**: 100 dispositivos IoT, 1 servidor IoT, 1 controlador SDN, 10 IoT *gateways* e 1 *switch* SDN.

Na Figura 7, apresentamos a arquitetura para a construção dos três cenários em análise. Observa-se que a imagem ilustra especificamente a configuração do primeiro cenário, no entanto, ressaltamos que essa representação serve como um modelo exemplar que abrange os demais cenários. A variação entre os cenários ocorre principalmente nos números associados aos dispositivos IoT e aos *gateways*, mantendo a estrutura arquitetônica geral consistente. Essa abordagem possibilita uma compreensão abrangente da base arquitetônica, enquanto as nuances específicas de cada cenário são facilmente discerníveis através das variações quantitativas destacadas.

Figura 7 – Arquitetura dos cenários.



Fonte: Elaborada pelo autor.

A ênfase na exploração de cenários em redes de menor porte decorreu da limitação dos recursos utilizados, os quais não possibilitavam a emulação de redes mais extensas. Além disso, essa abordagem foi adotada primariamente devido à necessidade de verificar, previamente, a funcionalidade do sistema desenvolvido, uma etapa que podia ser efetuada de maneira mais ágil e eficiente em ambientes de menor escala.

Os cenários de teste adotados para esta pesquisa são estruturados em redes que empregam os protocolos MQTT, responsável pela comunicação entre os sensores, e *IPv6 over Low power Wireless Personal Area Networks* (6LoWPAN). Nesse contexto, o foco dos ataques se

concentra no MQTT *broker*, que desempenha o papel de servidor IoT. Cada cenário é configurado com um controlador Ryu, no qual residem os módulos do sistema, e este está interligado ao *switch* SDN específico do ambiente. Essa arquitetura proporciona um ambiente propício para a avaliação da segurança nas redes IoT, destacando o papel crítico do MQTT *broker* como ponto de vulnerabilidade, enquanto o controlador Ryu e o *switch* SDN complementam a infraestrutura essencial para a execução dos testes.

Na implementação dos sensores, optamos por dois tipos distintos de troca de mensagens, *Time-Driven* e *Event-Driven*. O primeiro tipo refere-se aos sensores que enviam mensagens periodicamente em um intervalo pré-definido, exemplificado por sensores que transmitem informações de temperatura a cada minuto. O segundo tipo de sensor envia mensagens ao detectar a ocorrência de um evento específico, como nos sensores de movimento que comunicam informações sempre que identificam movimentação.

Em relação aos ataques, definimos dois tipos: SYN *Flood* e ICMP *Flood*. O primeiro consiste no envio de pacotes SYN para o dispositivo alvo em uma taxa elevada (MIRKOVIC; REIHER, 2004). O segundo envolve o envio de múltiplas mensagens ICMP *echo* (BHUYAN *et al.*, 2015). Nos experimentos, os ataques de cada serão tanto DoS como DDoS. Nos ataques de DoS, apenas um nó será utilizado como agente de ataque. Por outro lado, nos ataques de DDoS, a abordagem será mais intensiva, com 70% dos nós na rede desempenhando o papel de atacantes em cada cenário.

Para atingir os objetivos de validação, serão consideradas três métricas. A primeira verifica a alteração do valor de Hurst durante o ataque de negação de serviço comparado ao tráfego normal, podendo ser utilizado para detectar tais ataques. A segunda demonstra a eficiência do DDSHP na detecção, enquanto a terceira avalia o tempo necessário para detectar um ataque. A descrição de cada métrica segue abaixo:

a) **Hurst**: demonstrar a alteração do mesmo durante o ataque em relação ao padrão normal da rede;
b) **Matriz de confusão**: utilizada para determinar a quantidade correta de ataques (fluxos de ataque) detectados. A partir dessa matriz, serão calculadas a Acurácia, a Precisão e o *Recall*;
c) **Tempo de detecção**: indicar o intervalo decorrido entre o início do ataque e sua identificação.

Na seção subsequente, delinearemos os recursos e materiais empregados para a

concretização dos cenários previamente descritos. Detalharemos a escolha e utilização de cada componente, proporcionando uma visão abrangente do suporte empregado durante a implementação dos mencionados cenários.

6.2 Recursos

Para a construção dos cenários foi usado o *Mininet-WiFi*, com o qual é possível emular as redes de sensores nos seus diferentes tamanhos, além de permitir o uso da SDN. Os cenários foram executados em um *notebook* com sistema operacional Ubuntu 20.04, processador *Intel Core* i7, 16 gigabytes (GB) de armazenamento e 8 GB de memória *Random Access Memory* (RAM).

O controlador escolhido é o Ryu (RYU, 2022), desenvolvido em *Python*. Para facilitar a integração, os módulos do DDSHP também foram desenvolvidos nessa linguagem. Quanto aos módulos, o DM utiliza o *framework hurst* (MOTTL, 2023), desenvolvido em *Python*, para calcular H. Destacamos que, apesar do uso predominante de uma linguagem específica, o DDSHP pode ser adaptado para outras linguagens, desde que mantenha a lógica de desenvolvimento.

Desenvolvemos *scripts* em *Shell Script* para gerar tráfego normal nos sensores, simulando a comunicação com o servidor IoT (MQTT *Broker*) por meio de mensagens MQTT. As mensagens seguem os padrões *Time-Driven* (enviadas em intervalos predefinidos) e *Event-Driven* (enviadas ao ocorrer algum evento de interesse). Os padrões para cada sensor nos cenários de teste foram estabelecidos com base nos estudos de Li *et al.* (2020a), Yang *et al.* (2019), Wang *et al.* (2021) e Mocnej *et al.* (2018). A Tabela 4 apresenta os padrões de tráfego de cada sensor emulado.

Tabela 4 – Padrões de tráfego dos sensores dos cenários.

Sensor	**Frequência de envio (segundos)**
E-card check-in	Distribuição normal (média=20, variância=2)
Pagamento	Distribuição exponencial (média=2,5)
Fumaça	600 ou detectar fumaça
Luz	1800 ou detectar movimento
Umidade	1800
Temperatura	600
Vídeo de vigilância	15
Pressão sanguínea	0,5
Temperatura corporal	5
Medidor elétrico	600
Eletrocardiograma	0,25
Controle remoto	60 a 120

Fonte: Elaborada pelo autor.

Como foram elaborados três cenários distintos, cada um apresentando tamanhos e objetivos específicos. A composição de sensores para cada cenário foi selecionada levando em consideração as particularidades de cada ambiente. Dessa forma, garantimos que os sensores escolhidos estivessem alinhados às necessidades e características únicas de cada configuração. Sendo assim, essas na Tabela 5 exibimos os sensores que compunham os cenários:

Tabela 5 – Composição dos sensores de cada cenário.

Sensor	**Smart Home**	**Smart Market**	**Smart Hopsital**
E-card check-in		X	X
Pagamento		X	
Fumaça	X	X	X
Luz	X	X	X
Umidade	X		X
Temperatura	X	X	X
Vídeo de vigilância	X	X	X
Pressão sanguínea	X		X
Temperatura corporal			X
Medidor elétrico	X	X	X
Eletrocardiograma	X		X
Controle remoto	X	X	X

Fonte: Elaborada pelo autor.

Por fim, para simular tráfegos de ataque, desenvolvemos *scripts* também em *Shell Script*, em conjunto com a ferramenta SendIP (SENDIP, 2023), responsável pelo envio de pacotes. Esses *scripts* foram projetados para emular os ataques de negação de serviço. A combinação de *Shell Script* e SendIP proporciona a flexibilidade necessária para modelar cenários de ameaça realistas, permitindo a avaliação do sistema diante de potenciais ataques.

6.3 Resultados

Na presente seção, abordaremos individualmente cada experimento realizado, apresentando detalhes específicos adotados e os resultados correspondentes a cada um. A divisão permite uma análise mais aprofundada, oferecendo uma compreensão abrangente das descobertas e implicações derivadas de cada experimento.

Antes de discutirmos cada experimento, é necessário abordar as configurações específicas de cada um, incluindo métricas, fatores e níveis. Vale ressaltar que os fatores e níveis nos experimentos são similares, havendo pouca variação. A seguir, apresentam-se as configurações:

a) **Experimento 1:**

- **Métrica**: Parâmetro de Hurst;
- **Fatores e Níveis**:
  - Número de sensores: 20, 50 e 100;
  - Tipo de tráfego: Normal e ataque;
  - Tipo de ataque: DoS SYN, DDoS SYN, DoS ICMP e DDoS ICMP.

b) **Experimento 2:**

- **Métrica**: Matriz de confusão (acurácia, precisão e *recall*);
- **Fatores e Níveis**:
  - Número de sensores: 20, 50 e 100;
  - Tipo de tráfego: Normal e ataque.

c) **Experimento 3:**

- **Métrica**: Tempo de detecção;
- **Fatores e Níveis**:
  - Número de sensores: 20, 50 e 100;
  - Momento do ataque (em segundos): de 0 a 30 e de 31 a 60.

Após fornecer detalhes sobre as configurações de cada experimento, avançaremos para as próximas subseções, onde discutiremos os resultados obtidos em cada abordagem. Este exame permitirá uma compreensão das contribuições e implicações dos dados gerados, fundamentando as conclusões apresentadas neste estudo.

6.3.1 Experimento 1 (Estudo de Hurst)

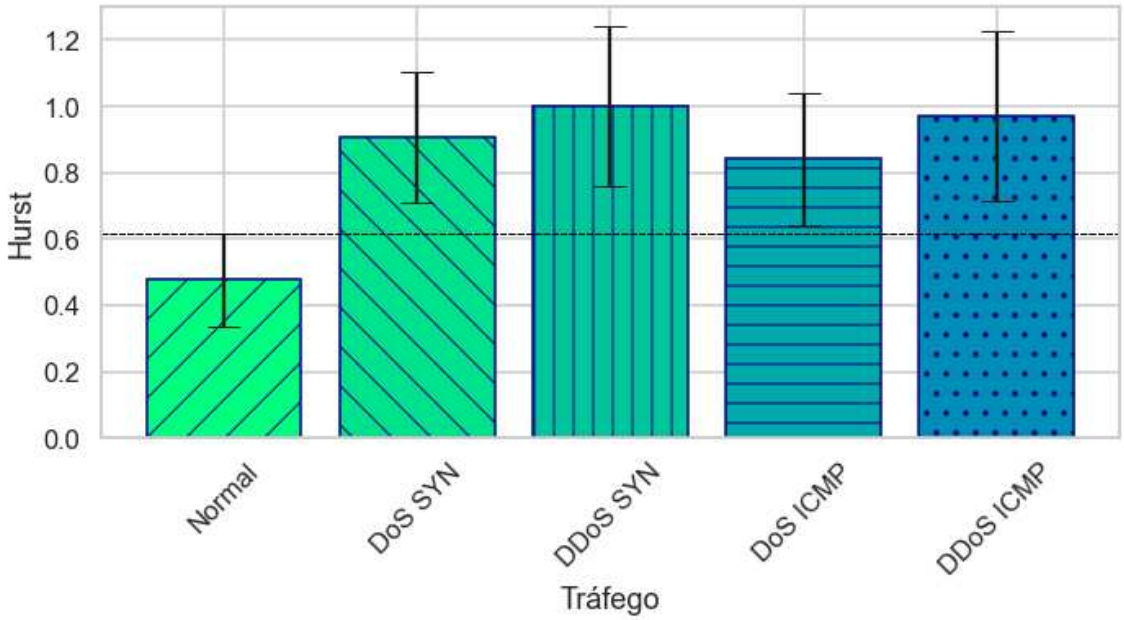
Neste experimento, a análise e interpretação dos dados serão conduzidas por meio da observação do comportamento do parâmetro de Hurst em duas situações distintas: sem a ocorrência de ataques e durante um ataque de negação de serviço. Essa abordagem visa identificar qualquer alteração significativa no padrão de Hurst durante a ocorrência de um ataque, permitindo uma compreensão mais aprofundada de como o sistema responde a esse tipo de evento. Ao comparar os resultados obtidos em ambas as situações, buscamos discernir indicativos claros de mudanças no Hurst.

Para gerar os dados do experimento, executamos os ambientes de teste por 2 horas, coletando o valor de Hurst a cada minuto para os tráfegos normais (sem ataque) e para cada tipo de ataque, DoS SYN, DDoS SYN, DoS ICMP e DDoS ICMP. Com os cálculos de Hurst realizados a cada minuto, foram coletadas mais de 120 amostras para cada cenário. Essas

amostras foram utilizadas para criar o intervalo de confiança para um nível de confiança de 99% do Hurst correspondente a cada tráfego analisado. O cálculo do Hurst foi realizado utilizando o DDSHP, que calcula e retorna o valor de Hurst a cada minuto.

Na Figura 8, apresentam-se os resultados do parâmetro de Hurst com nível de confinça de 99% para o cenário com 20 sensores (*Smart Home*). Os resultados revelam nuances distintas entre o tráfego normal e os ataques. No cenário de tráfego regular, o intervalo de confiança varia de 0,3347 a 0,6176, indicando uma persistência e um padrão de dependência de curto prazo. Por outro lado, nos momentos de ataques, observa-se uma ampliação da variabilidade e imprevisibilidade em comparação com o tráfego normal.

Figura 8 – Resultados do parâmetro de Hurst com nível de confiança de 99% para cada tráfego - Cenário *Smart Home* (20 sensores).



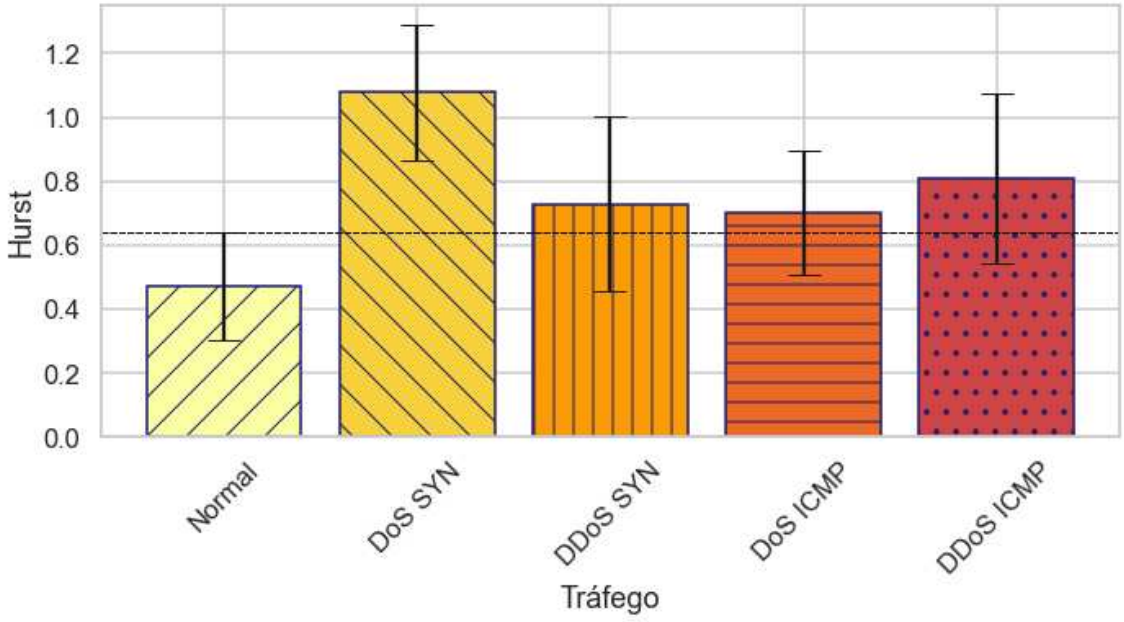
Fonte: Elaborada pelo autor.

Destaca-se que o intervalo de confiança do tráfego regular não se sobrepõe aos intervalos associados aos ataques, evidenciando uma alteração no Hurst durante a ocorrência desses eventos, ressaltando a utilidade do método na detecção e caracterização de mudanças no comportamento do sistema.

Na Figura 9, apresentam-se os resultados do parâmetro de Hurst com nível de confinça de 99% para o cenário com 50 sensores (*Smart Market*). Na análise dos resultados, observamos que, no tráfego normal, o intervalo de confiança (0,3018 a 0,6417) é ligeiramente mais amplo do que no cenário de 20 sensores, indicando uma maior variabilidade e menor persistência. Nos casos de tráfego durante ataques, essa imprevisibilidade e variabilidade

aumentam ainda mais em comparação com o cenário anterior.

Figura 9 – Resultados do parâmetro de Hurst com nível de confiança de 99% para cada tráfego - Cenário *Smart Market* (50 sensores).



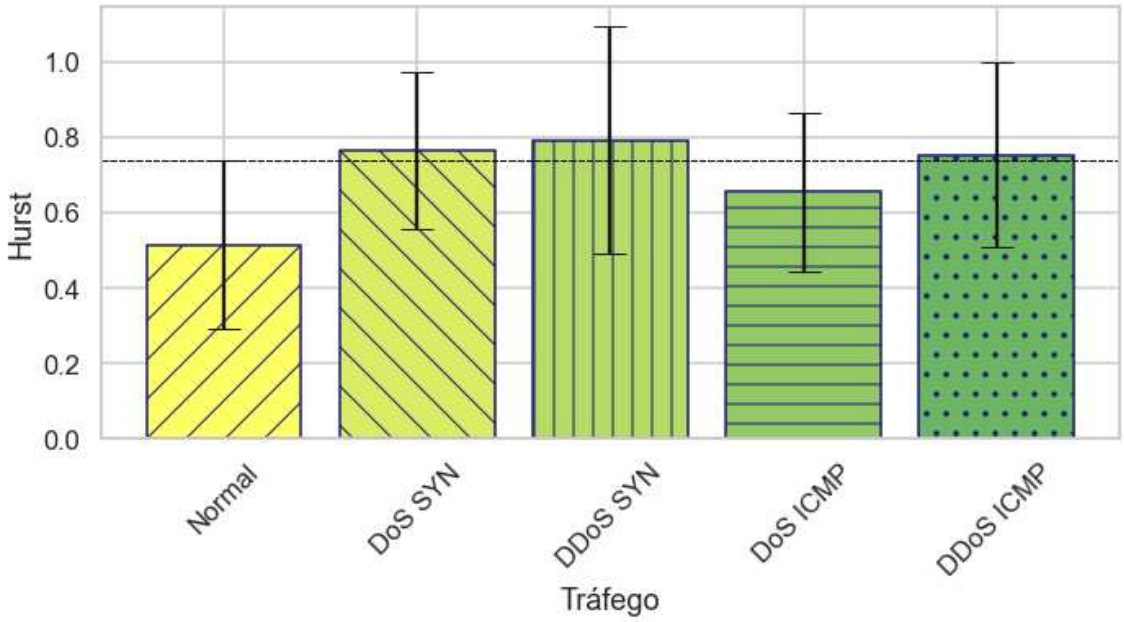
Fonte: Elaborada pelo autor.

O intervalo de confiança do tráfego normal tem intersecção com os intervalos de quase todos os ataques, exceto o DoS SYN, sugerindo que, em alguns momentos, o Hurst do tráfego de um ataque pode ser confundido como normal. No entanto, é importante destacar que, em média, os ataques apresentam um Hurst superior a 0,6, enquanto o tráfego normal permanece abaixo de 0,5. Essa diferença na média de Hurst reforça a capacidade de distinguir entre tráfego normal e ataques, mesmo quando há sobreposição em determinados momentos.

Na Figura 10, apresentam-se os resultados do parâmetro de Hurst com nível de confinça de 99% para o cenário com 100 sensores (*Smart Hospital*). Na análise dos resultados, observamos que, no tráfego normal, o intervalo de confiança abrange uma amplitude mais significativa, indo de 0,2906 a 0,7387, indicando maior variabilidade e baixa persistência. Nos tráfegos durante os ataques, a variabilidade se mantém similar à do tráfego normal.

O intervalo de confiança do tráfego normal tem intersecção com todos os intervalos dos ataques, evidenciando a variabilidade independente do tipo de tráfego. Entretanto, é possível discernir que os ataques introduzem alterações no Hurst. Como observado em cenários anteriores, a média de Hurst dos tráfegos durante os ataques apresenta alguma diferença, com um Hurst médio superior a 0,6, enquanto no tráfego normal, permanece em média de 0,5. Essa distinção ressalta o impacto dos ataques na dinâmica do tráfego.

Figura 10 – Resultados do parâmetro de Hurst com nível de confiança de 99% para cada tráfego - Cenário *Smart Hospital* (100 sensores).



Fonte: Elaborada pelo autor.

6.3.1.1 Considerações

Os resultados da análise revelam conclusões importantes sobre o comportamento do tráfego em diferentes cenários. Primeiramente, observamos que o tráfego normal demonstra uma certa persistência em cenários com um número reduzido de nós. Em contrapartida, os tráfegos de ataque exibem uma maior variabilidade e imprevisibilidade, independentemente do cenário. Além disso, constatamos que o número de nós exerce influência sobre a variabilidade do tráfego normal, manifestando-se por intervalos de confiança mais amplos à medida que o número de nós aumenta.

Em todos os cenários analisados, a média de Hurst dos ataques se manteve acima de 0,6, sendo notável que, mesmo no menor cenário, a média ultrapassou 0,8. Por outro lado, o tráfego normal apresentou uma média de Hurst em torno de 0,5. Essa disparidade revela que, de modo geral, há uma alteração perceptível no Hurst quando ocorrem ataques nos cenários analisados. Essa constatação sugere a viabilidade de desenvolver um algoritmo de detecção de ataques de negação de serviço, utilizando o Hurst como indicador-chave.

Destacamos que o primeiro experimento não apenas verifica a alteração no padrão de Hurst durante um ataque, mas também desempenha o papel de estudar a rede, fase essencial na configuração do DDSHP. Neste contexto, observamos que a definição da *baseline* no módulo de detecção do DDSHP não deve se basear apenas no intervalo de confiança, mas também na

média de Hurst.

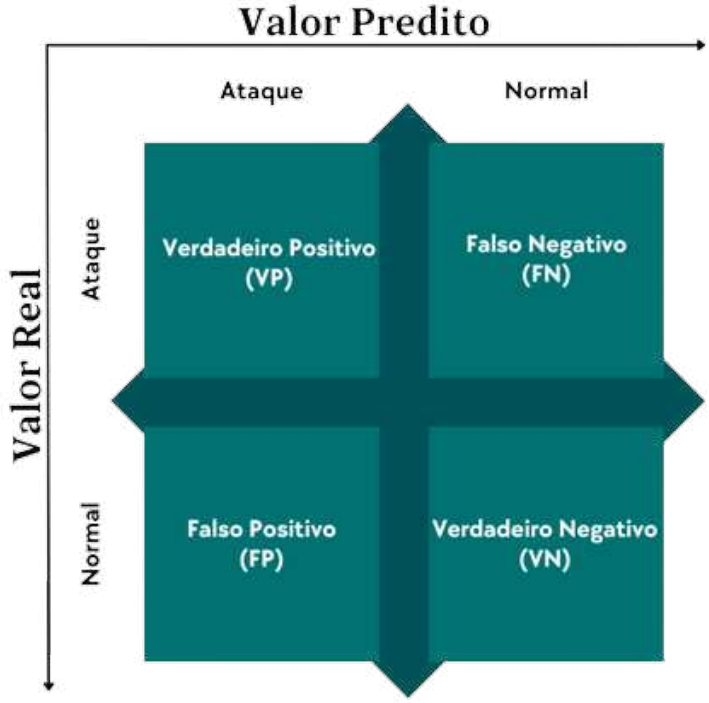
6.3.2 Experimento 2 (Matriz de Confusão)

No âmbito do segundo experimento, a opção recaiu sobre a utilização da matriz de confusão como instrumento na quantificação dos fluxos de ataque identificados. Através dessa matriz, procederemos aos cálculos, tais como acurácia, precisão (para a detecção de tráfego normal e de ataque) e recall (referente aos tráfegos normais e de ataques). Esta abordagem visa proporcionar uma análise do desempenho do sistema de detecção, contribuindo assim para a compreensão da eficácia do DDSHP.

A matriz de confusão utilizada para a classificação está exemplificada na Figura 11. Essa abordagem é frequentemente empregada em outros estudos, como demonstrado por Hafeez *et al.* (2020) e Ravi e Shalinie (2020). Portanto, a classificação adotada neste trabalho é a seguinte:

a) **Verdadeiro Positivo (VP)**: Tráfego de ataque classificado como ataque;
b) **Falso Positivo (FP)**: Tráfego normal classificado como ataque;
c) **Verdadeiro Negativo (VN)**: Tráfego normal classificado como normal;
d) **Falso Negativo (FN)**: Tráfego de ataque classificado como normal.

Figura 11 – Matriz de confusão para detecção de ataques.



Fonte: Elaborada pelo autor.

Nos experimentos conduzidos, geramos aproximadamente 5 mil fluxos de tráfegos para cada cenário, abrangendo uma diversidade que incluiu tanto padrões normais quanto ataques emulados. Essa variabilidade busca capturar nuances do ambiente investigado. O DDSHP teve a tarefa de classificar esses diversos fluxos. Essa escolha visa assegurar uma avaliação precisa e eficaz do desempenho do sistema diante de cenários diversos.

Para configurar adequadamente o sistema para este experimento, é imperativo realizar uma análise da rede na qual será aplicado, conforme já mencionado antes, considerando especialmente o intervalo de confiança do valor de Hurst. Com base nos resultados do experimento anterior, constatou-se que depender exclusivamente do intervalo de confiança para estabelecer os limiares (máximo e mínimo) de Hurst não é uma abordagem ideal. Isso se deve ao fato de que, na maioria dos cenários, os intervalos de confiança dos tráfegos normais e de ataque têm intersecção.

No entanto, uma observação foi a constante inferioridade da média de Hurst no tráfego normal, que permaneceu abaixo de 0,6 em todos os cenários, enquanto nos ataques ela se manteve acima desse valor. Diante disso, definiu-se que os limiares de valor de Hurst em cada cenário seriam estabelecidos com o limiar mínimo sendo o mesmo do intervalo de confiança e o limiar máximo fixado em 0,6, quando há intersecção. Essa configuração visa otimizar a sensibilidade do sistema diante de padrões distintos, proporcionando uma abordagem mais robusta na detecção de anomalias. Sendo assim, ficaram definidos os seguintes intervalos como padrão normal em cada cenário:

a) ***Smart Home*:** Como não houve intersecção dos intervalos, assim ficou definida a *baseline*:
   - **Hurst mínimo**: 0,33;
   - **Hurst mínimo**: 0,61.

b) ***Smart Market*:** Como houve intersecção dos intervalos, assim ficou definida a *baseline*:
   - **Hurst mínimo**: 0,30;
   - **Hurst mínimo**: 0,60.

c) ***Smart Hospital*:** Como houve intersecção dos intervalos, assim ficou definida a *baseline*:
   - **Hurst mínimo**: 0,29;
   - **Hurst máximo**: 0,60.

Após a realização dos testes e a coleta de dados inseridos na matriz de confusão, procedeu-se ao cálculo de cinco métricas para avaliar o desempenho do sistema na classificação do tráfego. Cada métrica desempenha um papel na compreensão das capacidades do sistema. Com isso, as métricas foram as seguintes:

a) **Acurácia:** denota quão bem o DDSHP identifica corretamente o tráfego da rede (normal ou ataque);

$$\frac{VP+VN}{VP+VN+FN+FP} \times 100. \tag{6.1}$$

b) **Precisão (Ataque):** denota com que precisão o DDSHP identifica ataques;

$$\frac{VP}{VP+FP} \times 100. \tag{6.2}$$

c) ***Recall* (Ataque):** denota quão bem o DDSHP identifica o ataque;

$$\frac{VP}{VP+FN} \times 100. \tag{6.3}$$

d) **Precisão (Normal):** denota com que precisão o DDSHP identifica o tráfego normal na rede;

$$\frac{VN}{VN+FP} \times 100. \tag{6.4}$$

e) ***Recall* (Normal):** indica quão bem o DDSHP identifica o tráfego normal na rede.

$$\frac{VN}{VN+FN} \times 100. \tag{6.5}$$

Na Tabela 6 temos os resultados do segundo experimento.

Tabela 6 – Resultados do Experimeto 2.

	Smart Home	***Smart Market***	***Smart Hospital***
Acurácia (%)	94,34	92,11	82,04
Precisão - Ataque (%)	98,22	94,75	86,35
Recall - Ataque (%)	90,78	89,91	81,01
Precisão - Normal (%)	98,21	94,53	83,37
Recall - Normal (%)	90,74	89,52	77,17

Fonte: Elaborada pelo autor.

Na análise de desempenho do modelo, observamos resultados distintos em diferentes cenários. Nos ambientes com 20 e 50 sensores (*Smart Home* e *Smart Market*), a acurácia demonstra-se notavelmente elevada, indicando que o modelo está acertando a maioria das

previsões, o que é promissor para contextos menos complexos. Entretanto, ao expandir o cenário para 100 nós (*Smart Hospital*), a acurácia registra uma redução para 82,04%. Essa diminuição sugere que o DDSHP pode encontrar dificuldades em lidar com cenários mais desafiadores e complexos, demandando uma análise mais aprofundada para compreender os fatores que impactam seu desempenho nesses contextos mais amplos.

A avaliação da precisão revela resultados interessantes em diferentes cenários. Em ambientes menos numerosos, a precisão na detecção de ataques é elevada, sugerindo que o DDSHP apresenta um desempenho consistente e preciso ao prever corretamente esses ataques. Em cenários mais desafiadores e extensos, como no *Smart Hospital*, há uma queda na precisão. Além disso, observou-se que a precisão na identificação do tráfego normal apresenta características semelhantes às da precisão na detecção de ataques, destacando a consistência do DDSHP em diferentes contextos de avaliação.

Já o *recall*, ao ser contrastado com a precisão, revela-se com menor porcentagem para identificar ataques, sugerindo que o modelo de detecção está deixando escapar mais instâncias de ataques, especialmente em cenários mais desafiadores. No entanto, este ainda se mostra com uma boa taxa de acertos em cenários menores. Além disso, nota-se uma tendência semelhante no *recall* para tráfego normal, indicando que o modelo pode também estar subestimando a proporção real de instâncias benignas. Essa análise aponta para a importância de aprimorar a sensibilidade do modelo, visando um desempenho mais robusto, principalmente em redes mais numerosas e complexas.

6.3.2.1 Considerações

Ao analisar o desempenho do DDSHP em redes IoT, observamos que ele demonstra uma solidez em ambientes mais restritos, porém enfrenta desafios à medida que o número de nós cresce. A acurácia, inicialmente estável, apresenta uma diminuição proporcional ao aumento do número de nós, indicando uma possível limitação do modelo em lidar com redes mais complexas. Tanto a precisão quanto o *recall* também mostram uma tendência de declínio, sugerindo que o algoritmo pode enfrentar dificuldades em manter um desempenho consistente à medida que a complexidade da rede se intensifica.

A redução no *recall* para ataques em cenários mais complexos destaca uma área de aprimoramento, especialmente se a detecção eficaz de ataques é uma prioridade. Em contextos de segurança IoT, onde a identificação abrangente de ameaças é crucial, a diminuição no *recall* pode

suscitar preocupações e apontar para a necessidade de ajustes visando aprimorar a capacidade do sistema em detectar ataques. Por outro lado, a alta precisão do algoritmo é digna de nota, indicando sua eficácia em evitar falsos positivos. Isso revela-se especialmente relevante em ambientes de recursos limitados, como as redes IoT, onde a otimização de recursos é essencial.

Em resumo, enquanto o DDSHP demonstra eficácia em cenários menos complexos, sugere-se que ajustes e otimizações sejam considerados para melhorar seu desempenho em redes IoT mais extensas. No entanto, destacamos que a facilidade de configuração do sistema torna esses ajustes viáveis e não representam obstáculos significativos.

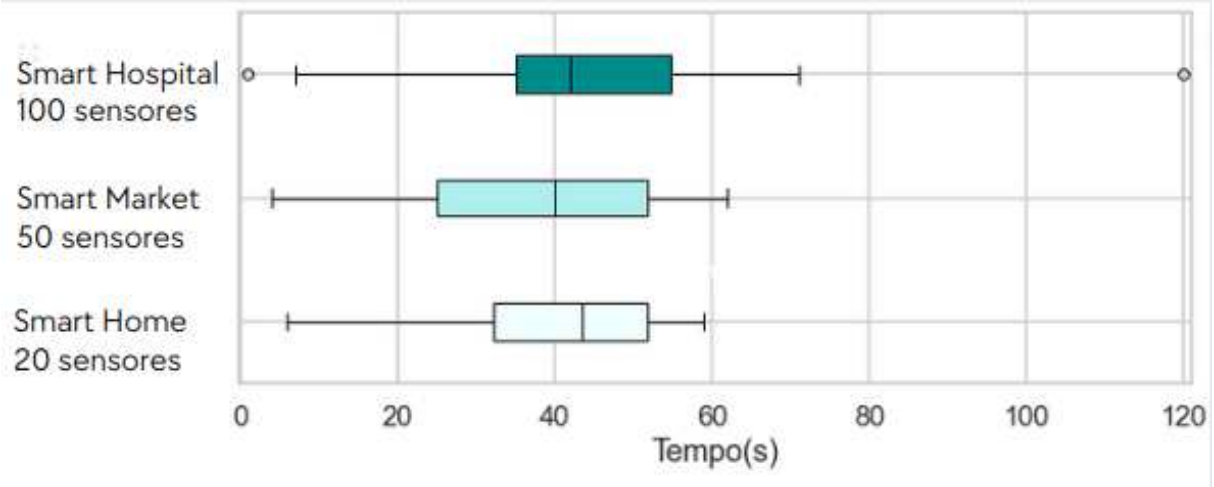
6.3.3 Experimento 3 (Tempo de Detecção)

Neste experimento, a principal análise concentra-se na avaliação do intervalo de tempo necessário para identificar um ataque desde o seu início. A observação desse período é relevante, uma vez que a prontidão na detecção do ataque é importante para notificar o administrador. Essa prontidão é essencial para prevenir a possível indisponibilidade da rede, proporcionando uma resposta proativa diante de ameaças potenciais. O estudo visa, assim, destacar a importância da eficiência na resposta de ataques.

No decorrer do experimento, executamos aproximadamente 100 ataques em cada cenário, os quais foram iniciados em períodos aleatórios de 0 a 30 segundos e de 31 a 60 segundos. A escolha desses intervalos se fundamenta no fato de que o DDSHP realiza seu cálculo de Hurst a cada minuto. Nesse contexto, torna-se importante avaliar a capacidade do sistema em detectar tais ataques dentro do prazo estipulado.

Na Figura 12, apresentam-se os resultados do terceiro experimento. No cenário com 20 sensores (*Smart Home*), o tempo mínimo registrado foi de 6 segundos, e o máximo, 59 segundos. No cenário com 50 sensores (*Smart Market*), os tempos mínimo e máximo foram 4 e 62 segundos, respectivamente. Já no cenário com 100 sensores (*Smart Hospital*), os tempos mínimo e máximo foram 1 e 120 segundos. Os resultados indicam que, na maioria das vezes, independentemente do cenário, o DDSHP leva menos de 1 minuto para detectar a ocorrência de ataques na rede. No entanto, mostra que pode levar mais minutos conforme se aumenta a complexidade da rede.

Figura 12 – Resultados do tempo de detecção.



Fonte: Elaborada pelo autor.

6.3.3.1 Considerações

Nossas conclusões revelam a eficácia do sistema no tempo de detecção de ataques, destacando-se pela agilidade ao realizar o cálculo, muitas vezes requerendo no máximo duas iterações para identificar a ameaça em questão. Em todos os cenários investigados, os tempos de detecção permaneceram consistentemente em uma faixa aceitável, geralmente ocorrendo em menos de um minuto, correspondente ao tempo de cada cálculo de Hurst.

No entanto, nota-se que o maior cenário estudado apresentou uma notável variação nos tempos de detecção, abrangendo desde 1 segundo até 120 segundos. Essa disparidade sugere desafios adicionais associados à detecção em redes de maior escala, como evidenciado de maneira congruente no experimento da matriz de confusão. Essa variação nos tempos de resposta pode indicar a necessidade de ajustes ou otimizações específicas ao lidar com redes mais extensas, visando manter a eficiência do sistema em diferentes contextos.

7 CONCLUSÕES E TRABALHOS FUTUROS

Visando avaliar a detecção de ataques de negação de serviço, focando em DDoS, em redes IoT, que possuem recursos limitados, desenvolvemos o DDSHP. Esse sistema utiliza o cálculo de Hurst para identificar possíveis ataques em uma rede. A integração da SDN na IoT foi realizada, pois o sistema utiliza recursos como o OpenFlow para coletar dados de tráfego da rede. O sistema é dividido em dois módulos: um responsável pela coleta de dados (CM) e outro pela detecção de ataques por meio do cálculo de Hurst (DM).

No decorrer do trabalho, exploramos os pilares deste estudo, que englobam a IoT, a SDN e o Parâmetro de Hurst. Além disso, analisamos as implicações de segurança na IoT e como a SDN e o Parâmetro de Hurst desempenham papéis nesse contexto, com uma atenção especial à segurança diante de ataques de DDoS.

Exploramos os trabalhos mais relevantes na literatura relacionados aos objetivos da presente pesquisa. Estruturamos a apresentação dos trabalhos em duas partes distintas: a primeira discorre sobre estudos que empregaram o Parâmetro de Hurst para a detecção de ataques de DDoS, oferecendo insights valiosos sobre essa abordagem específica. Por sua vez, a segunda parte concentra-se nas pesquisas que adotaram a SDN para fortalecer a segurança das redes IoT contra ataques de DDoS. Nessa segunda parte, observamos predominantemente a utilização de técnicas de aprendizado de máquina como estratégias para enfrentar os desafios inerentes à proteção das redes IoT contra ameaças dessa natureza.

O processo metodológico desta pesquisa percorreu diversas etapas para alcançar seus objetivos. Iniciou-se com uma pesquisa bibliográfica aprofundada, que fundamenta o conhecimento existente sobre o tema em questão. A etapa seguinte envolveu a definição do tema de concisamente, seguida pelo refinamento da proposta, onde foram delineados os contornos da pesquisa. Na definição dos experimentos estabelecemos a estrutura experimental necessária para a coleta de dados. A construção do cenário, por sua vez, proporcionou desenvolver o ambiente controlado para a execução dos experimentos. No desenvolvimento do sistema as ideias teóricas foram transformadas em implementações práticas. Os testes funcionais validaram o funcionamento efetivo do sistema, preparando o terreno para a etapa subsequente de avaliação de desempenho. Esta fase analisou a eficácia do sistema em relação aos critérios estabelecidos. Finalmente, a análise e apresentação dos resultados encerram o ciclo metodológico, oferecendo uma síntese interpretativa e conclusiva das descobertas alcançadas ao longo da pesquisa.

Os experimentos conduzidos tinham como objetivo inicial verificar a viabilidade do

uso do cálculo de Hurst na detecção de ataques de negação de serviço (DoS e DDoS) e avaliar o desempenho do DDSHP. Os resultados revelaram que o sistema apresenta bom desempenho em redes menores, tanto na classificação quanto no tempo de identificação dos ataques. No entanto, em redes maiores e mais complexas, o DDSHP demonstrou deficiências.

Salientamos que a impossibilidade de reduzir os intervalos de confiança para obter uma maior diferenciação entre o tráfego real e o de ataque se deve principalmente à constatação de que aumentar o tamanho da amostra acarretaria um aumento no atraso na detecção. A busca por uma diferenciação mais nítida entre os padrões de tráfego legítimo e as atividades maliciosas são um objetivo essencial, porém, a ampliação da amostra, embora possa aprimorar a precisão estatística, inevitavelmente resultaria em uma demora adicional na identificação de possíveis ameaças. Assim, a decisão de manter os intervalos de confiança em um determinado patamar visa equilibrar a sensibilidade da detecção com a necessidade de uma resposta rápida e eficiente diante de potenciais ataques, considerando a complexidade e as limitações inerentes ao ambiente monitorado.

A principal contribuição deste trabalho foi o desenvolvimento do sistema DDSHP, que se mostrou consistente na detecção de ataques de negação de serviço em redes menores, o que o torna apto para ser aplicado neste tipo de rede. Além disso, o estudo evidenciou que o cálculo de Hurst é de fato alterado em ataques desse tipo.

No âmbito do desenvolvimento futuro do DDSHP, identificam-se algumas áreas passíveis de aprimoramento e exploração. Primeiramente, buscaremos otimizar a eficácia do sistema em redes mais extensas, concentrando nossos esforços na melhoria do seu poder de identificação de tráfegos de ataque. Outro ponto a ser abordado é o desenvolvimento de um módulo de mitigação automático. Essa adição ao sistema permitirá que o processo de eliminação de ameaças ocorra de forma autônoma, dispensando a intervenção direta do administrador da rede e proporcionando respostas mais rápidas diante de potenciais ataques.

Para aprimorar a sensibilidade e eficiência do DDSHP, pretendemos reduzir o intervalo de tempo de cálculo do Hurst, passando de uma periodicidade de um minuto para 30 segundos. Essa medida visa tornar o sistema mais ágil na detecção de padrões e na adaptação a mudanças no tráfego da rede. Adicionalmente, pretendemos conduzir uma análise comparativa entre o DDSHP e soluções similares já desenvolvidas. Esse exercício permitirá destacar o desempenho do nosso sistema em relação às alternativas existentes.

Finalmente, para validar ainda mais a eficácia do DDSHP, planejamos realizar testes

em ambientes reais. Essa etapa serve para verificar como o sistema se comporta em situações do mundo real, proporcionando *insights* valiosos sobre seu desempenho prático e sua capacidade de resposta em cenários dinâmicos.

REFERÊNCIAS

ABBOU, A. N.; BADDI, Y.; HASBI, A. Software Defined Networks in Internet of Things Integration Security: Challenges and Solutions. In: **2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)**. [*S. l.*: *s. n.*], 2018. p. 1–6.

AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. **IEEE Communications Surveys & Tutorials**, v. 17, n. 4, p. 2347–2376, 2015. ISSN 1553-877X, 2373-745X. Disponível em: https://ieeexplore.ieee.org/document/7123563/.

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. **Computer Networks**, v. 54, n. 15, p. 2787–2805, out. 2010. ISSN 13891286. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S1389128610001568.

BANERJEE, M.; SAMANTARAY, S. D. Network Traffic Analysis Based IoT Botnet Detection Using Honeynet Data Applying Classification Techniques. **International Journal of Computer Science and Information Security (IJCSIS)**, v. 17, n. 8, p. 61–66, 2019.

BARSUKOV, I. S.; BOBRESHOV, A. M.; RIAPOLOV, M. P. Fractal Analysis based Detection of DoS/LDoS Network Attacks. In: **2019 International Russian Automation Conference (RusAutoCon)**. Sochi, Russia: IEEE, 2019. p. 1–5. ISBN 9781728102658. Disponível em: https://ieeexplore.ieee.org/document/8867618/.

BERA, S.; MISRA, S.; VASILAKOS, A. V. Software-Defined Networking for Internet of Things: A Survey. **IEEE Internet of Things Journal**, v. 4, n. 6, p. 1994–2008, dez. 2017. ISSN 2327-4662. Disponível em: http://ieeexplore.ieee.org/document/8017556/.

BHUYAN, M. H.; BHATTACHARYYA, D.; KALITA, J. An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection. **Pattern Recognition Letters**, v. 51, p. 1–7, 2015. ISSN 0167-8655. Disponível em: https://www.sciencedirect.com/science/article/pii/S016786551400244X.

BULL, P.; AUSTIN, R.; POPOV, E.; SHARMA, M.; WATSON, R. Flow Based Security for IoT Devices Using an SDN Gateway. In: **2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)**. Vienna, Austria: IEEE, 2016. p. 157–163. ISBN 9781509040520. Disponível em: http://ieeexplore.ieee.org/document/7575858/.

CHEN, W.; XIAO, S.; LIU, L.; JIANG, X.; TANG, Z. A DDoS attacks traceback scheme for SDN-based smart city. **Computers & Electrical Engineering**, v. 81, p. 106503, jan. 2020. ISSN 00457906. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S004579061832901X.

CHEN, Y.-W.; SHEU, J.-P.; KUO, Y.-C.; CUONG, N. V. Design and Implementation of IoT DDoS Attacks Detection System based on Machine Learning. In: **2020 European Conference on Networks and Communications (EuCNC)**. Dubrovnik, Croatia: IEEE, 2020. p. 122–127. ISBN 9781728143552. Disponível em: https://ieeexplore.ieee.org/document/9200909/.

CHENG, H.; LIU, J.; XU, T.; REN, B.; MAO, J.; ZHANG, W. Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks. **International Journal of Sensor Networks**, v. 34, n. 1, p. 56, 2020. ISSN 1748-1279, 1748-1287. Disponível em: http://www.inderscience.com/link.php?id=109720.

CROVELLA, M.; BESTAVROS, A. Self-similarity in World Wide Web traffic: evidence and possible causes. **IEEE/ACM Transactions on Networking**, v. 5, n. 6, p. 835–846, dez. 1997. ISSN 10636692. Disponível em: http://ieeexplore.ieee.org/document/650143/.

DEKA, R. K.; BHATTACHARYYA, D. K. Self-similarity based ddos attack detection using hurst parameter. **Security and Communication Networks**, Wiley Online Library, v. 9, n. 17, p. 4468–4481, 2016.

DILLON, C.; BERKELAAR, M. **OpenFlow (D)DoS Mitigation**. [*S. l.*], 2014. 17 p. Disponível em: https://www.os3.nl/_media/2013-2014/courses/rp1/p42_report.pdf.

DJOUANI, R.; DJOUANI, K.; BOUTEKKOUK, F.; SAHBI, R. A Security Proposal for IoT integrated with SDN and Cloud. In: **2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)**. [*S. l.*: *s. n.*], 2018. p. 1–5.

DONNO, M. D.; TANGE, K.; DRAGONI, N. Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog. **IEEE Access**, v. 7, p. 150936–150948, 2019.

FAJAR, A. P.; PURBOYO, T. W. A Survey Paper of Distributed Denial-of-Service Attack in Software Defined Networking (SDN). **International Journal of Applied Engineering Research (IJAER)**, v. 13, n. 1, p. 476–482, 2018. Disponível em: https://www.ripublication.com/ijaer18/ijaerv13n1_64.pdf.

FAN, T.; CHEN, Y. A scheme of data management in the Internet of Things. In: **2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content**. [*S. l.*: *s. n.*], 2010. p. 110–114.

FARRIS, I.; TALEB, T.; KHETTAB, Y.; SONG, J. A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems. **IEEE Communications Surveys & Tutorials**, v. 21, n. 1, p. 812–837, 2019. ISSN 1553-877X, 2373-745X. Disponível em: https://ieeexplore.ieee.org/document/8424018/.

FONTES, R. R.; AFZAL, S.; BRITO, S. H. B.; SANTOS, M. A. S.; ROTHENBERG, C. E. Mininet-WiFi: Emulating software-defined wireless networks. In: **2015 11th International Conference on Network and Service Management (CNSM)**. Barcelona, Spain: IEEE, 2015. p. 384–389. ISBN 9783901882777. Disponível em: http://ieeexplore.ieee.org/document/7367387/.

GALEANO-BRAJONES, J.; CARMONA-MURILLO, J.; VALENZUELA-VALDéS, J. F.; LUNA-VALERO, F. Detection and Mitigation of DoS and DDoS Attacks in IoT-Based Stateful SDN: An Experimental Approach. **Sensors**, v. 20, n. 3, p. 816, jan. 2020. Disponível em: https://www.mdpi.com/1424-8220/20/3/816.

GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of Things (IoT): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, v. 29, n. 7, p. 1645–1660, set. 2013. ISSN 0167739X. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0167739X13000241.

HAFEEZ, I.; ANTIKAINEN, M.; DING, A. Y.; TARKOMA, S. IoT-KEEPER: Detecting Malicious IoT Network Activity Using Online Traffic Analysis at the Edge. **IEEE Transactions on Network and Service Management**, v. 17, n. 1, p. 45–59, mar. 2020. ISSN 1932-4537, 2373-7379. Disponível em: https://ieeexplore.ieee.org/document/8960276/.

HAYAJNEH, A. A.; BHUIYAN, M. Z. A.; MCANDREW, I. Improving Internet of Things (IoT) Security with Software-Defined Networking (SDN). **Computers**, v. 9, n. 1, p. 8, fev. 2020. ISSN 2073-431X. Disponível em: https://www.mdpi.com/2073-431X/9/1/8.

HUANG, Y.; LI, G. Descriptive models for Internet of Things. In: **2010 International Conference on Intelligent Control and Information Processing**. [*S. l.*: *s. n.*], 2010. p. 483–486.

IDRIS, F.; HAMEED, S. Software Defined Security Service Provisioning Framework for Internet of Things. **International Journal of Advanced Computer Science and Applications**, v. 7, n. 12, 2016. ISSN 21565570, 2158107X. Disponível em: http://thesai.org/Publications/ViewPaper?Volume=7&Issue=12&Code=ijacsa&SerialNo=54.

IHS Markit. **8 in 2018: The top transformative technologies to watch this year**. Londres, 2018. 16 p. Disponível em: https://www.forbes.com/sites/louiscolumbus/2018/12/13/2018-roundup-of-internet-of-things-forecasts-and-market-estimates/?sh=71d7d6b97d83.

IOULIANOU, P.; VASILAKIS, V.; MOSCHOLIOS, I.; LOGOTHETIS, M. A Signature-based Intrusion Detection System for the Internet of Things. In: **Information and Communication Technology Form**. AUT: York, 2018. Disponível em: https://eprints.whiterose.ac.uk/133312/.

ITU. **Recommendation ITU-T Y.2060: Overview of the Internet of things**. [*S. l.*]: International Telecommunication Union, 2012.

JEONG, H.-D. J.; AHN, W.; KIM, H.; LEE, J.-S. R. Anomalous Traffic Detection and Self-Similarity Analysis in the Environment of ATMSim. **Cryptography**, v. 1, n. 3, p. 24, dez. 2017. Disponível em: https://www.mdpi.com/2410-387X/1/3/24.

KALKAN, K.; ZEADALLY, S. Securing Internet of Things with Software Defined Networking. **IEEE Communications Magazine**, v. 56, n. 9, p. 186–192, set. 2018. ISSN 0163-6804, 1558-1896. Disponível em: https://ieeexplore.ieee.org/document/8121868/.

KANAGAVELU, R.; AUNG, K. M. M. A Survey on SDN Based Security in Internet of Things. In: ARAI, K.; KAPOOR, S.; BHATIA, R. (Ed.). **Advances in Information and Communication Networks**. Cham: Springer International Publishing, 2019. v. 887, p. 563–577. ISBN 9783030034047 9783030034054. Disponível em: http://link.springer.com/10.1007/978-3-030-03405-4_39.

KARAARSLAN, E.; KARABACAK, E.; CETINKAYA, C. Design and Implementation of SDN-Based Secure Architecture for IoT-Lab. In: HEMANTH, D. J.; KOSE, U. (Ed.). **Artificial Intelligence and Applied Mathematics in Engineering Problems**. Cham: Springer International Publishing, 2020. v. 43, p. 877–885. ISBN 9783030361778 9783030361785. Disponível em: http://link.springer.com/10.1007/978-3-030-36178-5_76.

KAUR, G.; SAXENA, V.; GUPTA, J. Detection of TCP targeted high bandwidth attacks using self-similarity. **Journal of King Saud University - Computer and Information Sciences**, v. 32, n. 1, p. 35–49, jan. 2020. ISSN 13191578. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S1319157817300617.

KHAN, R.; KHAN, S. U.; ZAHEER, R.; KHAN, S. Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges. In: **2012 10th International Conference on Frontiers of Information Technology**. [*S. l.*: *s. n.*], 2012. p. 257–260.

KOTENKO, I.; SAENKO, I.; LAUTA, O.; KRIBEL, A. An Approach to Detecting Cyber Attacks against Smart Power Grids Based on the Analysis of Network Traffic Self-Similarity. **Energies**, v. 13, n. 19, p. 5031, set. 2020. ISSN 1996-1073. Disponível em: https://www.mdpi.com/1996-1073/13/19/5031.

KREUTZ, D.; RAMOS, F. M. V.; VERISSIMO, P. E.; ROTHENBERG, C. E.; AZODOLMOLKY, S.; UHLIG, S. Software-Defined Networking: A Comprehensive Survey. **Proceedings of the IEEE**, v. 103, n. 1, p. 14–76, jan. 2015. ISSN 0018-9219, 1558-2256. Disponível em: http://ieeexplore.ieee.org/document/6994333/.

KRISHNAN, P.; NAJEEM, J. S.; ACHUTHAN, K. SDN Framework for Securing IoT Networks. In: KUMAR, N.; THAKRE, A. (Ed.). **Ubiquitous Communications and Network Computing**. Cham: Springer International Publishing, 2018. v. 218, p. 116–129. ISBN 9783319734224 9783319734231. Disponível em: http://link.springer.com/10.1007/978-3-319-73423-1_11.

Lei Zhang; Jiping Zhu; Hai Huang. An SDN_based management framework for IoT devices. In: **25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communities Technologies (ISSC 2014/CIICT 2014)**. Limerick, Ireland: Institution of Engineering and Technology, 2014. p. 175–179. ISBN 9781849199247. Disponível em: https://digital-library.theiet.org/content/conferences/10.1049/cp.2014.0680.

LELAND, W. E.; TAQQU, M. S.; WILLINGER, W.; WILSON, D. V. On the self-similar nature of ethernet traffic (extended version). **IEEE/ACM Transactions on networking**, IEEE, v. 2, n. 1, p. 1–15, 1994.

LI, J.; ALTMAN, E.; TOUATI, C. A General SDN-based IoT Framework with NVF Implementation. **ZTE Communications**, ZTE Corporation, v. 13, n. 3, p. 42–45, set. 2015. Disponível em: https://hal.inria.fr/hal-01197042.

LI, M. An approach to reliably identifying signs of DDOS flood attacks based on LRD traffic pattern recognition. **Computers & Security**, v. 23, n. 7, p. 549–558, out. 2004. ISSN 01674048. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0167404804001245.

LI, Y.; JIN, D.; WANG, B.; SU, X.; RIEKKI, J.; SUN, C.; WEI, H.; WANG, H.; HAN, L. Predicting internet of things data traffic through lstm and autoregressive spectrum analysis. In: **NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium**. [*S. l.*: *s. n.*], 2020. p. 1–8.

LI, Z.; XING, W.; KHAMAISEH, S.; XU, D. Detecting Saturation Attacks Based on Self-Similarity of OpenFlow Traffic. **IEEE Transactions on Network and Service Management**, v. 17, n. 1, p. 607–621, mar. 2020. ISSN 1932-4537, 2373-7379. Disponível em: https://ieeexplore.ieee.org/document/8932535/.

LOKSHINA, I.; ZHONG, H.; LANTING, C. J. M. Self-similar Teletraffic in a Smart World. In: KRYVINSKA, N.; GREGUš, M. (Ed.). **Data-Centric Business and Applications**. Cham: Springer International Publishing, 2020. v. 30, p. 137–160. ISBN 9783030190682 9783030190699. Disponível em: http://link.springer.com/10.1007/978-3-030-19069-9_5.

LYSENKO, S.; BOBROVNIKOVA, K.; MATIUKH, S.; HURMAN, I.; SAVENKO, O. Detection of the botnets' low-rate DDoS attacks based on self-similarity. **International Journal of Electrical and Computer Engineering (IJECE)**, v. 10, n. 4, p. 3651, ago. 2020. ISSN 2722-2578, 2088-8708. Disponível em: http://ijece.iaescore.com/index.php/IJECE/article/view/20780.

MA, D.; XU, Z.; LIN, D. Defending Blind DDoS Attack on SDN Based on Moving Target Defense. In: TIAN, J.; JING, J.; SRIVATSA, M. (Ed.). **International Conference on Security and Privacy in Communication Networks**. Cham: Springer International Publishing, 2015. v. 152, p. 463–480. ISBN 9783319238289 9783319238296. Disponível em: http://link.springer.com/10.1007/978-3-319-23829-6_32.

MARTINEZ-JULIA, P.; SKARMETA, A. Empowering the Internet of Things with Software Defined Networking [WHITE PAPER]. In: . [*S. l.*: *s. n.*], 2014.

MARZANO, A.; ALEXANDER, D.; FONSECA, O.; FAZZION, E.; HOEPERS, C.; STEDING-JESSEN, K.; CHAVES, M. H. P. C.; CUNHA, I.; GUEDES, D.; MEIRA, W. The Evolution of Bashlite and Mirai IoT Botnets. In: **2018 IEEE Symposium on Computers and Communications (ISCC)**. Natal: IEEE, 2018. p. 00813–00818. ISBN 9781538669501. Disponível em: https://ieeexplore.ieee.org/document/8538636/.

MIORANDI, D.; SICARI, S.; PELLEGRINI, F. D.; CHLAMTAC, I. Internet of things: Vision, applications and research challenges. **Ad hoc networks**, Elsevier, v. 10, n. 7, p. 1497–1516, 2012.

MIRKOVIC, J.; REIHER, P. A taxonomy of ddos attack and ddos defense mechanisms. **SIGCOMM Comput. Commun. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 34, n. 2, p. 39–53, apr 2004. ISSN 0146-4833. Disponível em: https://doi.org/10.1145/997150.997156.

MOCNEJ, J.; PEKAR, A.; SEAH, W. K.; ZOLOTOVA, I. Network traffic characteristics of the iot application use cases. In: . [*S. n.*], 2018. Disponível em: https://api.semanticscholar.org/CorpusID:198181520.

MOTTL, D. **hurst 0.0.5**. 2023. https://pypi.org/project/hurst. 10 de Novembro, 2023. Disponível em: {https://pypi.org/project/hurst}.

ONF. **Open Networking Foundation (ONF)**. 2021. https://opennetworking.org/. 4 de Junho, 2021. Disponível em: {https://opennetworking.org}.

PA, Y. M. P.; SUZUKI, S.; YOSHIOKA, K.; MATSUMOTO, T.; KASAMA, T.; ROSSOW, C. IoTPOT: Analysing the Rise of IoT Compromises. In: **9th USENIX Workshop on Offensive Technologies (WOOT 15)**. Washington, D.C.: USENIX Association, 2015. Disponível em: https://www.usenix.org/conference/woot15/workshop-program/presentation/pa.

PACHECO, L. A. B.; GONDIM, J. J. C.; BARRETO, P. A. S.; ALCHIERI, E. Evaluation of Distributed Denial of Service threat in the Internet of Things. In: **2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)**. Cambridge, Boston, MA, USA: IEEE, 2016. p. 89–92. ISBN 9781509032167. Disponível em: http://ieeexplore.ieee.org/document/7778599/.

PATIL, G.; MCCLEAN, S.; RAINA, G. Drop tail and red queue management with small buffers: stability and hopf bifurcation. **ICTACT Journal on Communication Technology**, v. 2, n. 2, p. 339–344, 2011.

PAXSON, V.; FLOYD, S. Wide area traffic: the failure of Poisson modeling. **IEEE/ACM Transactions on Networking**, v. 3, n. 3, p. 226–244, jun. 1995. ISSN 10636692. Disponível em: http://ieeexplore.ieee.org/document/392383/.

RAFIQUE, W.; HE, X.; LIU, Z.; SUN, Y.; DOU, W. CFADefense: A Security Solution to Detect and Mitigate Crossfire Attacks in Software-Defined IoT-Edge Infrastructure. In: **2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)**. Zhangjiajie, China: IEEE, 2019. p. 500–509. ISBN 9781728120584. Disponível em: https://ieeexplore.ieee.org/document/8855406/.

RAVI, N.; SHALINIE, S. M. Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture. **IEEE Internet of Things Journal**, v. 7, n. 4, p. 3559–3570, abr. 2020. ISSN 2327-4662, 2372-2541. Disponível em: https://ieeexplore.ieee.org/document/8993716/.

RYU. **Ryu SDN Framework**. 2022. https://ryu-sdn.org/. 4 de Março, 2022. Disponível em: {https://ryu-sdn.org}.

SENDIP. **hurst 0.0.5**. 2023. https://www-x.antd.nist.gov/ipv6/sendip.html. 10 de Novembro, 2023. Disponível em: {https://www-x.antd.nist.gov/ipv6/sendip.htm}.

SETHI, P.; SARANGI, S. R. Internet of Things: Architectures, Protocols, and Applications. **Journal of Electrical and Computer Engineering**, v. 2017, p. 1–25, 2017. ISSN 2090-0147, 2090-0155. Disponível em: https://www.hindawi.com/journals/jece/2017/9324035/.

SHIN, S.; XU, L.; HONG, S.; GU, G. Enhancing Network Security through Software Defined Networking (SDN). In: **2016 25th International Conference on Computer Communication and Networks (ICCCN)**. Waikoloa, HI, USA: IEEE, 2016. p. 1–9. ISBN 9781509022793. Disponível em: https://ieeexplore.ieee.org/document/7568520/.

SHIRALI-SHAHREZA, S.; GANJALI, Y. FleXam: flexible sampling extension for monitoring and security applications in openflow. In: **Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking - HotSDN '13**. Hong Kong, China: ACM Press, 2013. p. 167. ISBN 9781450321785. Disponível em: http://dl.acm.org/citation.cfm?doid=2491185.2491215.

SILVA, F. S. D.; SILVA, E.; NETO, E. P.; LEMOS, M.; NETO, A. J. V.; ESPOSITO, F. A Taxonomy of DDoS Attack Mitigation Approaches Featured by SDN Technologies in IoT Scenarios. **Sensors**, v. 20, n. 11, p. 3078, maio 2020. ISSN 1424-8220. Disponível em: https://www.mdpi.com/1424-8220/20/11/3078.

SILVEIRA, F. A. F.; LIMA-FILHO, F.; SILVA, F. S. D.; JUNIOR, A. de M. B.; SILVEIRA, L. F. Smart Detection-IoT: A DDoS Sensor System for Internet of Things. In: **2020 International Conference on Systems, Signals and Image Processing (IWSSIP)**. Niterói, Brazil: IEEE, 2020. p. 343–348. ISBN 9781728175393. Disponível em: https://ieeexplore.ieee.org/document/9145265/.

TAHAEI, H.; AFIFI, F.; ASEMI, A.; ZAKI, F.; ANUAR, N. B. The rise of traffic classification in IoT networks: A survey. **Journal of Network and Computer Applications**, v. 154, p. 102538, mar. 2020. ISSN 10848045. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S1084804520300126.

TAYYABA, S. K.; SHAH, M. A.; KHAN, O. A.; AHMED, A. W. Software Defined Network (SDN) Based Internet of Things (IoT): A Road Ahead. In: **Proceedings of the International Conference on Future Networks and Distributed Systems**. New York, NY, USA: Association

for Computing Machinery, 2017. (ICFNDS '17). ISBN 9781450348447. Disponível em: https://doi.org/10.1145/3102304.3102319.

TSYBAKOV, B.; GEORGANAS, N. D. Self-similar traffic and upper bounds to buffer-overflow probability in an ATM queue. **Performance Evaluation**, v. 32, n. 1, p. 57–80, fev. 1998. ISSN 01665316. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0166531697000163.

VILALTA, R.; CIUNGU, R.; MAYORAL, A.; CASELLAS, R.; MARTINEZ, R.; PUBILL, D.; SERRA, J.; MUNOZ, R.; VERIKOUKIS, C. Improving Security in Internet of Things with Software Defined Networking. In: **2016 IEEE Global Communications Conference (GLOBECOM)**. [*S. l.*: *s. n.*], 2016. p. 1–6.

VISHWAKARMA, R.; JAIN, A. K. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. **Telecommunication Systems**, v. 73, n. 1, p. 3–25, jan. 2020. ISSN 1018-4864, 1572-9451. Disponível em: http://link.springer.com/10.1007/s11235-019-00599-z.

WANG, S.; GOMEZ, K.; SITHAMPARANATHAN, K.; ASGHAR, M. R.; RUSSELLO, G.; ZANNA, P. Mitigating DDoS Attacks in SDN-Based IoT Networks Leveraging Secure Control and Data Plane Algorithm. **Applied Sciences**, v. 11, n. 3, p. 929, jan. 2021. ISSN 2076-3417. Disponível em: https://www.mdpi.com/2076-3417/11/3/929.

WANI, A.; REVATHI, S. DDoS Detection and Alleviation in IoT using SDN (SDIoT-DDoS-DA). **Journal of The Institution of Engineers (India): Series B**, v. 101, n. 2, p. 117–128, abr. 2020. ISSN 2250-2106, 2250-2114. Disponível em: https://link.springer.com/10.1007/s40031-020-00442-z.

YAN, Q.; HUANG, W.; LUO, X.; GONG, Q.; YU, F. R. A Multi-Level DDoS Mitigation Framework for the Industrial Internet of Things. **IEEE Communications Magazine**, v. 56, n. 2, p. 30–36, fev. 2018. ISSN 0163-6804. Disponível em: http://ieeexplore.ieee.org/document/8291111/.

YANG, Y.; WANG, J.; ZHAI, B.; LIU, J. Iot-based ddos attack detection and mitigation using the edge of sdn. In: VAIDYA, J.; ZHANG, X.; LI, J. (Ed.). **Cyberspace Safety and Security**. Cham: Springer International Publishing, 2019. p. 3–17. ISBN 978-3-030-37352-8.

YIN, D.; ZHANG, L.; YANG, K. A DDoS Attack Detection and Mitigation With Software-Defined Internet of Things Framework. **IEEE Access**, v. 6, p. 24694–24705, 2018. ISSN 2169-3536. Disponível em: https://ieeexplore.ieee.org/document/8352645/.

YOUSUF, O.; MIR, R. N. A Survey on Security Enhancements in the Internet of Things Using Software-Defined-Networking (SDN). **International Journal of Computing and Digital Systems**, v. 9, n. 4, p. 591–606, jul. 2020. ISSN 2210-142X. Disponível em: https://journal.uob.edu.bh:443/handle/123456789/3884.

YU, S. J.; KOH, P.; KWON, H.; KIM, D. S.; KIM, H. K. Hurst Parameter based Anomaly Detection for Intrusion Detection System. In: **2016 IEEE International Conference on Computer and Information Technology (CIT)**. Nadi, Fiji: [*S. n.*], 2016. p. 234–240.

YU, T.; SEKAR, V.; SESHAN, S.; AGARWAL, Y.; XU, C. Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things. In: **Proceedings of the 14th ACM Workshop on Hot Topics in Networks**. Philadelphia PA USA: ACM, 2015. p. 1–7. ISBN 9781450340472. Disponível em: https://dl.acm.org/doi/10.1145/2834050.2834095.

ZHANG, C.; GREEN, R. Communication security in internet of thing: preventive measure and avoid DDoS attack over IoT network. In: **Proceedings of the 18th Symposium on Communications & Networking**. Alexandria, Virginia: Society for Computer Simulation International, 2015. (CNS '15), p. 8–15. ISBN 9781510801004.