UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

DIOGO VINÍCIUS MARTINS DA CRUZ

IDENTIFICAÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO DISTRIBUÍDO NA COMPUTAÇÃO EM NÉVOA UTILIZANDO UM SISTEMA DE INFERÊNCIA FUZZY

DISSERTAÇÃO

CURITIBA
2021

DIOGO VINÍCIUS MARTINS DA CRUZ

IDENTIFICAÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO DISTRIBUÍDO NA COMPUTAÇÃO EM NÉVOA UTILIZANDO UM SISTEMA DE INFERÊNCIA FUZZY

Identification of Distributed Denial of Service Attacks in Fog Computing using a Fuzzy Inference System.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná como requisito parcial para a obtenção do título de "Mestre em Computação Aplicada" - Área de Concentração: Engenharia de Sistemas Computacionais.

Orientadora: Dra. Ana Cristina Barreiras Kochem Vendramin
Coorientador: Dr. Daniel Fernando Pigatto

CURITIBA

2021

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba

DIOGO VINICIUS MARTINS DA CRUZ

IDENTIFICAÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO DISTRIBUÍDO NA COMPUTAÇÃO EM NÉVOA UTILIZANDO UM SISTEMA DE INFERÊNCIA FUZZY

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Computação Aplicada da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Sistemas Computacionais.

Data de aprovação: 30 de Abril de 2021

Prof.a Ana Cristina Barreiras Kochem Vendramin, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Carlos Alberto Maziero, Doutorado - Universidade Federal do Paraná (Ufpr)

Prof Luiz Nacamura Junior, Doutorado - Universidade Tecnológica Federal do Paraná

Prof.a Myriam Regattieri De Biase Da Silva Delgado, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 30/04/2021.

Dedico este trabalho aos meus antigos (eternos)
professores, que doaram-se à arte de ensinar,
fazendo-me amar esta que é mais que uma
profissão, é um sacerdócio

AGRADECIMENTOS

Este trabalho é fruto de muito esforço e dedicação, entretanto, sua conclusão não seria possível sem, primeiramente, a graça e a misericórdia de Deus, que fez com que meus objetivos fossem alcançados, permitindo-me superar limitações e ultrapassar todos os obstáculos encontrados.

Certamente, não será possível atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já, peço desculpas àquelas que não estão presentes entre estas palavras, entretanto, tenham certeza de que contam com as minhas orações e eterna gratidão.

À minha esposa Ana Caroline, pelo incentivo irrestrito, tolerância nos momentos difíceis e compreensão em períodos de ausência. Alguém que abdicou de suas prioridades em detrimento da realização deste sonho, que é nosso. Sua presença ao meu lado desde antes do início desta caminhada fortaleceu meus passos em busca da vitória.

À minha filha Laura, que ainda no ventre materno, enriqueceu seu pai com força e esperança de que, em um futuro próximo, esta também será a sua vitória.

Aos meus orientadores, que, de maneira sobremodo excelente, depositaram confiança, paciência e dedicação na concretização deste sonho, sendo assertivos em todas as suas intervenções. Pessoas com as quais pude contar independente do dia e horário, contribuindo com observações pontuais, as quais me conduziram no caminho correto.

A todos os professores e colegas da Universidade Tecnológica Federal do Paraná, que contribuíram de maneira significativa na conclusão deste trabalho.

Aos amigos de trabalho, superiores, pares e subordinados, que atuaram com bondade, compreensão e camaradagem com seu irmão de farda.

Enfim, a todos os que de alguma forma contribuíram de maneira direta ou indireta para a realização de mais este sonho.

Um ponto de encontro onde não mais apenas "usaremos um computador", mas onde o "computador se use" independentemente, de modo a tornar a vida mais eficiente. Os objetos – as "coisas" – estarão conectados entre si e em rede, de modo inteligente, e passarão a "sentir" o mundo ao redor e a interagir (ASHTON, KEVIN, 2015).

RESUMO

CRUZ, Diogo Vinícius Martins da Cruz. **Identificação de Ataques de Negação de Serviço Distribuído na Computação em Névoa utilizando um Sistema de Inferência Fuzzy**. 56 f. Dissertação de Mestrado - Programa de Pós-Graduação em Computação Aplicada, Universidade Tecnológica Federal do Paraná. Curitiba - PR, 2021.

A constante utilização de dispositivos de Internet das Coisas (*Internet of Things* - IoT) eleva a preocupação com a segurança cibernética devido a limitação de recursos desses dispositivos. A Computação em Névoa, camada intermediária entre a nuvem e a IoT, possui diversas vulnerabilidades conhecidas, as quais são exploradas por ameaças como o Ataque de Negação de Serviço Distribuído (*Distributed Denial of Service* - DDoS). O MQTT (*Message Queue Telemetry Transport*) é o protocolo mais utilizado entre a névoa e a IoT. Na camada da névoa, um *broker* MQTT é o equipamento responsável pelo gerenciamento das comunicações oriundas dos dispositivos de IoT, sendo um dos nós mais visados em ataques DDoS. Esta modalidade de ataque, ao esgotar os recursos físicos do *broker*, pode representar desde um pequeno atraso nas comunicações, até a interrupção total do serviço. Este trabalho propõe um Sistema de Inferência Fuzzy (SIF) capaz de detectar, identificar e inferir o grau de pertinência de um ataque DDoS em um nó de névoa por meio da análise do seu padrão de consumo energético.

Palavras-chave: Computação em Névoa. Lógica Fuzzy. Ataque de Negação de Serviço Distribuído. Consumo Energético.

ABSTRACT

CRUZ, Diogo Vinícius Martins da Cruz. **Identification of Distributed Denial of Service Attacks in Fog Computing using a Fuzzy Inference System**. 56 f. Dissertação de Mestrado - Programa de Pós-Graduação em Computação Aplicada, Universidade Tecnológica Federal do Paraná. Curitiba - PR, 2021.

The constant use of Internet of Things (IoT) devices raises concern about cybersecurity due to limited resource of these devices. Fog Computing, an intermediate layer between cloud and IoT, has several known vulnerabilities, which can be exploited by threats such as the Distributed Denial of Service (DDoS) attack. MQTT (Message Queue Telemetry Transport) is the most widely used protocol between fog computing and IoT. In the fog computing, an MQTT broker is the equipment responsible for managing communications originating from IoT devices, being one of the most targeted device in DDoS attacks. This attack, when depleting the broker's physical resources, can cause different levels of problems ranging from a small delay in communications, to a complete interruption of service. This work proposes a Fuzzy Inference System (FIS) capable of detecting, identifying and infer the degree of membership of a DDoS attack in a fog node by analyzing its energy consumption pattern.

Keywords: Fog Computing. Fuzzy Logic. Distributed Denial of Service Attack. Energy Consumption.

LISTA DE ALGORITMOS

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS E ACRÔNIMOS

AMQP	*Advanced Message Queuing Protocol*
CA	Consumo Alto
CB	Consumo Baixo
CEA	Consumo Médio Alto
CM	Consumo Médio
CMA	Consumo Muito Alto
CoAP	*Constrained Application Protocol*
DDoS	*Distributed Denial of Service*
DDS	*Data Distribution Service*
DoS	*Denial of Service*
HTTP	*Hypertext Transfer Protocol*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IoT	*Internet of Things*
IPG	Inundação por Pacotes Grandes
IPv6	*Internet Protocol - version 6*
LAN	*Local Area Network*
M2M	*Machine-to-Machine*
MIM	*Man-in-the-Middle*
MQTT	*Message Queue Telemetry Transport*
NA	Não Ataque
OSI	*Open System Interconnection*
QoS	*Quality Of Service*
RFID	*Radio Frequency Identification*
SDN	*Software Defined Networking*
SIF	Sistema de Inferência *Fuzzy*
SSL/TLS	*Secure Sockets Layer/Transport Layer Security*
TC	Tempo Curto
TCP/IP	*Transmission Control Protocol/Internet Protocol*
TL	Tempo Longo
TM	Tempo Médio
TMC	Tempo Muito Curto
TML	Tempo Muito Longo

SUMÁRIO

1 INTRODUÇÃO

Equipamentos com inteligência funcional vêm ocupando atualmente lacunas importantes no dia a dia das pessoas, influenciando comportamentos e proporcionando melhoria na qualidade de vida (FIRDOUS *et al.*, 2018). Este cenário de computação ubíqua, conhecido como Internet das Coisas ou *Internet of Things* (IoT), favorece o crescimento de adoções de medidas de segurança. Os estudos relacionados à identificação, prevenção e mitigação de ameaças em IoT têm crescido nos últimos anos (SIKORA *et al.*, 2019; VISHWAKARMA; JAIN, 2020; XIAO *et al.*, 2019). O ambiente de IoT está hierarquicamente dividido em camadas nas quais grande parte de suas soluções de mitigação de ataques são destinadas apenas aos níveis superiores desta arquitetura, não contemplando igualmente as camadas inferiores (BUTUN *et al.*, 2020). Devido às restrições de recursos físicos em dispositivos de IoT, o controle de consumo de energia elétrica vem incorporando as medidas de segurança neste ecossistema (FIRDOUS *et al.*, 2018). Estima-se um crescimento de cerca de 75 bilhões de dispositivos de IoT conectados até 2025 (BUTUN *et al.*, 2020). Este aumento no número de dispositivos conectados cria um gargalo nos enlaces de comunicação, devido à grande quantidade de mensagens trocadas entre os nós. Uma medida introduzida pela Cisco para minimizar o problema de latência na rede de comunicação foi a criação da Computação em Névoa (*Fog Computing*), uma camada de análise de dados localizada entre a nuvem e o limite da rede (dispositivos de IoT) (PERALTA *et al.*, 2017; PAN *et al.*, 2020). A camada de névoa permite a coexistência de diferentes protocolos de comunicação, como o *Constrained Application Protocol* (CoAP), o *Data Distribution Service* (DDS), o *Hypertext Transfer Protocol* (HTTP) e o *Advanced Message Queuing Protocol* (AMQP) (NAZIR; KALEEM, 2019; POTRINO *et al.*, 2019; KEPCEOGLU *et al.*, 2019). Entretanto, o *Message Queue Telemetry Transport* (MQTT) figura como um protocolo amplamente usado nas interações entre os dispositivos de IoT e computação em névoa (TRUONG; L., 219; ANDY *et al.*, 2017).

O Ataque de Negação de Serviço Distribuído ou *Distributed Denial of Service* (DDoS), é a ameaça mais observada em ambientes de névoa que utilizam o protocolo MQTT, principalmente quando a modalidade utilizada é o ataque volumétrico (*Volume-based*), também chamado de Ataque de Inundação (*Flooding Attack*) (VISHWAKARMA; JAIN, 2020). O modelo de ameaças da camada de névoa define as inundações por pacotes grandes, pacotes *Internet Control Message Protocol* (ICMP) e pacotes SYN como os tipos de DDoS mais presentes neste ambiente.

Estes tipos de ataques visam drenar os recursos físicos dos nós de névoa através da inundação de pacotes na rede, forçando-os a um elevado gasto energético (ROOHI *et al.*, 2019). Os ataques volumétricos quando direcionados à computação em névoa, nos casos em que esta se utiliza do protocolo MQTT, têm como alvo principal os dispositivos centrais de sua arquitetura, os *brokers* MQTT. Tais dispositivos são classificados como nós de névoa (*Fog Nodes*) (HARIPRIYA; KULOTHUNGAN, 2019). Apesar de serem geralmente mais robustos que os dispositivos de borda (ou seja, os dispositivos IoT), os *brokers* têm funções essenciais e a desativação ou interrupção temporária desses nós afeta amplamente os dispositivos a eles conectados (VISHWAKARMA; JAIN, 2020).

A detecção de um ataque com base na análise do tráfego da rede é algo muito comum na literatura. Porém, este tipo de análise nem sempre é uma estratégia viável, pois, por vezes, é difícil distinguir entre um tráfego normal e um tráfego de ataque (SIKORA *et al.*, 2019). Existem categorias de ataque onde as mudanças no tráfego da rede são mínimas, o que pode gerar falsos negativos (TIAN, 2020). Em outros casos, nós legítimos de IoT podem gerar padrões de tráfego incomuns. Com isso, considerar apenas este tipo de dado pode levar a um aumento de falsos positivos na detecção de ataques de inundação (ROOHI *et al.*, 2019). Embora algumas soluções de detecção sejam baseadas em *logs* do sistema, esses dados podem ser facilmente falsificados ou sua análise negligenciada (LI *et al.*, 2019).

O presente trabalho se propõe a detectar e identificar um ataque DDoS e eliminar uma série de lacunas nos modelos mencionados acima, pois indica a incidência de um ataque a partir da análise do padrão de consumo de energia elétrica da vítima, neste caso, um *broker* MQTT. Com isso, obtém-se uma solução viável para a detecção e identificação de ataques volumétricos na computação em névoa, informando, ainda, a força com que cada um desses ataques atinge o alvo, através da análise de seus respectivos graus de pertinência, implementado via uso da lógica *fuzzy* (ZADEH, 1975). A lógica *fuzzy* foi usada por identificar mais claramente cada tipo de ataque DDoS, em comparação com a lógica binária convencional, e por fornecer seus respectivos graus de pertinência. Ela modela sistemas não lineares e complexos (semelhante ao raciocínio humano), diminuindo a dependência de modelos matemáticos complexos (SHAH, 2018; BEROUINE *et al.*, 2019). A detecção de um ataque de maneira tempestiva favorece a criação de contramedidas imediatas para impedir ou minimizar seus efeitos, ainda, sua correta identificação auxilia na tomada de decisões precisas e mais efetivas na mitigação desta ameaça.

1.1 CONTEXTUALIZAÇÃO

A grande importância de um *broker* no gerenciamento de mensagens entre as camadas de IoT e névoa, torna-o alvo de uma gama de ameaças cibernéticas, dentre elas o ataque DDoS, o qual visa drenar seus recursos de *hardware* e gerar indisponibilidade de serviços (FIRDOUS *et al.*, 2018; DEOGIRIKAR; VIDHATE, 2017).

Quando o protocolo MQTT é utilizado em comunicações IoT, a camada de névoa é o local de atuação do *broker*, sendo este o nó de névoa responsável pelo tratamento das informações. Os dispositivos de IoT, em sua maioria, são responsáveis por tratar o tráfego de dados sensíveis, muitas vezes confidenciais e que requerem alta disponibilidade (VISHWAKARMA; JAIN, 2020; KORONIOTIS *et al.*, 2019). Eles são equipamentos que normalmente possuem poucos recursos de *hardware* e, com isso, são mais afetados por ataques de características físicas (FIRDOUS *et al.*, 2018; KEPCEOGLU *et al.*, 2019). As limitações do protocolo MQTT, a grande demanda no tráfego de dados em IoT, as vulnerabilidades inerentes a esta tecnologia e a escassez de recursos em seus ativos corroboraram com o crescimento de ataques cibernéticos neste ambiente, os quais visam drenar seus recursos físicos, obter acesso a informações sigilosas, gerar indisponibilidade de serviço, entre outros (FIRDOUS *et al.*, 2018; DEOGIRIKAR; VIDHATE, 2017).

1.2 OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS

O objetivo geral deste trabalho é propor um novo método, baseado em lógica *fuzzy*, de detecção e identificação de ataques de DDoS, utilizando como métrica o consumo energético de um *broker* MQTT em relação ao tempo. Os ataques que possuem características volumétricas (inundação) deixam um rastro energético bem característico, possibilitando assim inferir o tipo desse ataque e o seu grau de pertinência.

O detalhamento do objetivo geral é apresentado na forma dos seguintes objetivos específicos:

- Colher e analisar os dados de consumo energético de um *broker* MQTT em situação normal de operação e sob diferentes tipos de ataques de inundação;
- Submeter os dados colhidos a um Sistema de Inferência *Fuzzy*;
- Identificar o tipo de ataque e inferir o grau de pertinência com que este incide no *broker*.

1.3 CONTRIBUIÇÕES

Até o presente momento não foram encontradas pesquisas que propõem avaliar o comportamento do consumo energético de um nó de nevoa e, a partir dele e da lógica *fuzzy*, detectar, identificar e inferir o grau de pertinência com que diferentes tipos de ataques de negação de serviço atingem o alvo.

Os resultados iniciais deste trabalho, onde procurou-se identificar apenas a ocorrência de um ataque DDoS em um *broker* através da lógica *fuzzy*, sem identificar seu tipo, gerou a seguinte publicação:

CRUZ, D. V. M.; PIGATTO, D. F.; VENDRAMIN, A. C. B. K. Análise de Consumo Energético e Identificação de Ataques de Negação de Serviço em Computação em Névoa. In: XVIII Workshop em Clouds e Aplicações, Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2020, Rio de Janeiro. 38º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2020. p. 1-14.

1.4 ESTRUTURA DO DOCUMENTO

Este documento está organizado em cinco capítulos. O Capítulo 1 apresenta a introdução, contextualização, objetivos e contribuições deste trabalho. O Capítulo 2 apresenta a fundamentação teórica que descreve com mais detalhes a Internet das Coisas, a Computação em Névoa, o protocolo MQTT e a Lógica *Fuzzy*, e os trabalhos relacionados. O Capítulo 3 apresenta a arquitetura proposta para medir o consumo de energia de um nó de névoa e identificar ataques DDoS nesse nó. No Capítulo 4 são apresentados os resultados obtidos. Finalizando, o Capítulo 5 traz as considerações finais e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Este capítulo apresenta em detalhes os conceitos relacionados a este trabalho: Internet das Coisas e Computação em Névoa (Seção 2.1), Protocolo MQTT (Seção 2.2) e Lógica *Fuzzy* (Seção 2.3). Ao final do capítulo, alguns trabalhos relacionados são descritos (Seção 2.4).

2.1 INTERNET DAS COISAS E COMPUTAÇÃO EM NÉVOA

O termo IoT (*Internet of Things*) foi primeiramente proposto por Kevin Ashton no final dos anos 1990 e pode ser definido como um ambiente de objetos inteligentes conectados, os quais interagem com o mundo real (KEVIN, 2010; HANDA, 2018). A IoT explora as características comuns desses objetos, transformando-os em "coisas" inteligentes que são virtualmente identificadas e usam interfaces inteligentes para conectar-se em contextos sociais (SINGH; DEEPAK, 2019; ZHAO, 2014).

A intensa interação entre coisas e pessoas, proporcionada pela adoção da IoT, influencia no comportamento e qualidade de vida de seus usuários. Contudo, a interação entre equipamentos inteligentes, conhecida como *Machine-to-Machine* (M2M), também pode ocorrer com pouca ou mesmo nenhuma intervenção humana (FIRDOUS *et al.*, 2018; SINGH; DEEPAK, 2019).

A divisão funcional em camadas, conforme observada no modelo *Open System Interconnection* (OSI), é adotada também pela IoT, que usa uma divisão de tarefas hierarquicamente organizada. Cada uma dessas camadas tem suas vulnerabilidades conhecidas, as quais podem ser exploradas por cibercriminosos (LI *et al.*, 2019; VISHWAKARMA; JAIN, 2020; LIN *et al.*, 2017; HANDA, 2018).

A arquitetura de IoT é mais comumente apresentada em três camadas: Camada de Percepção, Camada de Aplicação e Camada de Rede (XU *et al.*, 2019; PIERLEONI *et al.*, 2020; ALABA *et al.*, 2017). A camada de percepção (também conhecida como Camada de Sensor) é composta por componentes físicos de IoT, como sensores e dispositivos atuadores de *Radio Frequency Identification* (RFID), e é responsável pela captura de dados por meio de radiofrequências (XU *et al.*, 2019; ACETO *et al.*, 2019). A Camada de Rede é responsável pela intercomunicação e roteamento dos dados coletados pelos sensores e é composta por tecnologias como Wi-Fi, Zigbee e 4G. A Camada de Aplicação, composta por diferentes protocolos (como, por exemplo, o MQTT), é onde são desenvolvidas a inteligência e as regras de negócio da arquitetura, tornando

os dados oriundos dos sensores úteis aos usuários finais (MISHRA; KERTESZ, 2020; KHANAM *et al.*, 2020; PIERLEONI *et al.*, 2020)

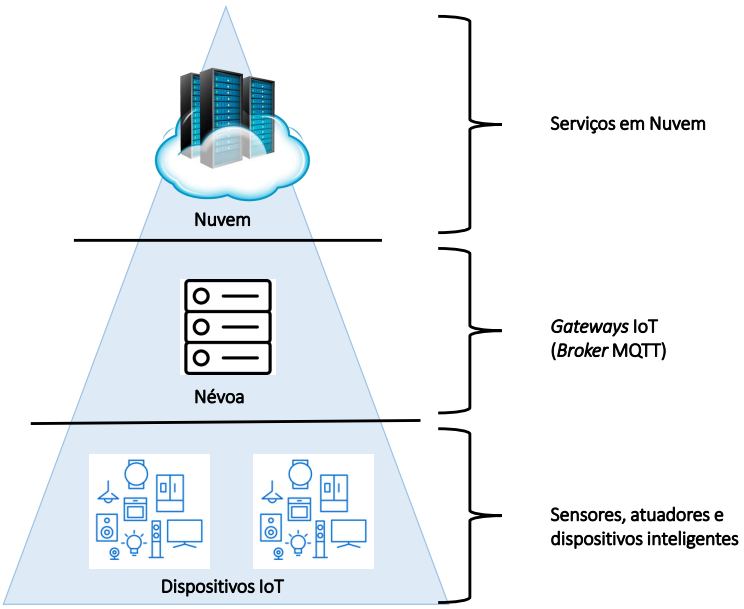
Uma das principais características dos dispositivos IoT é a limitação de recursos. Redes compostas por esses dispositivos normalmente contêm sensores e atuadores que são pouco robustos estruturalmente. Esta característica apresenta vantagens como economia de energia e baixo custo, mas é amplamente explorada por atacantes (FIRDOUS *et al.*, 2018; DIRO *et al.*, 2020).

A IoT será impulsionada pela implementação completa do *Internet Protocol - version 6* (IPv6), devido à distribuição de bilhões de endereços únicos, permitindo uma identificação exclusiva para cada um dos dispositivos de IoT. Estima-se que existirão 29,3 bilhões de dispositivos de IoT na rede em 2023, um aumento de 62% dos 18,4 bilhões registrados em 2018. Com base nesses dados, a comunicação M2M será composta por 50% de toda a demanda da rede nos próximos três anos (CISCO, 2020). Este crescimento constante e o conjunto heterogêneo de soluções de *hardware* e *software* que compõem o ambiente de IoT favorecem o surgimento de novas vulnerabilidades, tornando-o suscetível a ataques (KEPCEOGLU *et al.*, 2019). Devido às limitações de recursos dos sensores de borda e ao grande volume de dados a serem analisados, é necessária a adoção de uma infraestrutura que processe esses dados e trate as informações sensíveis de maneira eficiente. Nesse contexto, surgiu a Computação em Névoa. (BONOMI *et al.*, 2012; ATLAM *et al.*, 2018).

A Figura 1 apresenta uma arquitetura em três camadas, contemplando IoT, névoa e nuvem. A Computação em Névoa integra a borda da rede (dispositivos de IoT) ao seu núcleo (computação em nuvem), consistindo em uma camada intermediária, sendo considerada uma extensão da nuvem (ABEDI; POURKIANI, 2020; OSANAIYE *et al.*, 2017; MISHRA; KERTESZ, 2020). Os dispositivos finais, localizados no nível mais baixo da arquitetura e, portanto, os mais numerosos, comunicam-se com os *gateways* IoT, que operam na camada de névoa. A camada de névoa proporciona uma maior flexibilidade e escalabilidade às redes IoT, permitindo uma comunicação com o nível mais alto onde se encontram os servidores da nuvem (PERALTA *et al.*, 2017; EMA *et al.*, 2019; De Donno *et al.*, 2019).

O surgimento da computação em névoa se deve, em parte, à falta de serviços de qualidade nas fronteiras da rede para lidar com uma grande quantidade de dados de fontes distribuídas geograficamente (OSANAIYE *et al.*, 2017). Portanto, a camada de névoa aproxima a computação em nuvem dos dispositivos finais de IoT (dispositivos de borda), realizando o

Figura 1 – Arquitetura em Três Camadas: IoT, Névoa e Nuvem



Fonte: Adaptado de Peralta *et al.* (2017)

processamento e a análise dos dados mais próximos da origem, permitindo que qualquer ação derivada seja executada mais rapidamente (XU *et al.*, 2019; ABEDI; POURKIANI, 2020; EMA *et al.*, 2019; De Donno *et al.*, 2019).

A Computação em névoa atua em uma arquitetura composta por três camadas. A camada mais externa, a da Internet das Coisas, é composta por "coisas" inteligentes, como sensores, atuadores e sistemas embarcados. A camada intermediária consiste na névoa propriamente dita, composta por dispositivos destinados à análise e processamento de dados, como os *brokers*. A camada de nuvem é o nível mais alto desta arquitetura, sendo formada por servidores robustos que possuem grande poder computacional (DIRO *et al.*, 2020; PERALTA *et al.*, 2017; MISHRA; KERTESZ, 2020).

Existem vários protocolos de camada de aplicação usados nas interações entre as camadas de névoa e IoT. Os protocolos mais comuns são: o MQTT, o CoAP, o DDS, o HTTP e o AMQP (DIRO *et al.*, 2020; NAZIR; KALEEM, 2019; POTRINO *et al.*, 2019; KEPCEOGLU *et al.*, 2019; MISHRA; KERTESZ, 2020). Neste artigo o protocolo escolhido foi o MQTT, por ser amplamente utilizado nas comunicações entre IoT e névoa, principalmente quando dispositivos com poucos recursos computacionais são empregados (por exemplo, atuadores e sensores).

2.2 PROTOCOLO MQTT

O MQTT é um protocolo leve de troca de mensagens, que atua na Camada de Aplicação (modelo OSI) e é amplamente utilizado na computação em névoa (ANDY *et al.*, 2017; FIRDOUS *et al.*, 2018). O protocolo MQTT foi projetado para redes com baixa largura de banda, garantindo um certo grau de precisão atuando sob a pilha de protocolos *Transmission Control Protocol/Internet Protocol* (TCP/IP). O número da porta padrão do MQTT é a 18883/TCP para mensagens regulares ou a 8883/TCP usando criptografia *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) (TRUONG; L., 219; FIRDOUS *et al.*, 2018; LEKIĆ *et al.*, 2019)

Em uma comunicação através do MQTT, há um componente central chamado *broker*. Este componente atua na camada de névoa, sendo classificado como nó de névoa. Nós de névoa são infraestruturas de *hardware* que fornecem recursos e serviços ao limite da rede (JUTADHAMAKORN *et al.*, 2018; DIRO *et al.*, 2020; XU *et al.*, 2019; MISHRA; KERTESZ, 2020). O *broker* é responsável por receber todas as mensagens, filtrá-las e determinar quais dispositivos devem recebê-las (JUTADHAMAKORN *et al.*, 2018; G. GEORGE COULOURIS, JEAN DOLLIMORE, TIM KINDBERG, 2012). Um *broker* também pode disponibilizar uma camada de segurança simples para troca de mensagens, feita por meio de autenticação do usuário.

Em um ambiente que utiliza o protocolo MQTT, não é possível para um cliente encaminhar mensagens diretamente para outro cliente sem passar pelo *broker*. Isto se deve à adoção do paradigma Publicação e Assinatura (*Publish and Subscribe*) (FIRDOUS *et al.*, 2018; PERALTA *et al.*, 2017). O paradigma publicação e assinatura é uma alternativa ao modelo cliente/servidor tradicional, observado no protocolo HTTP [1]. No modelo cliente/servidor, um dispositivo pode enviar mensagens diretamente ao cliente final. Isso não é possível no paradigma Publicação/Assinatura, onde, independentemente do protocolo que está sendo implementado, as duas extremidades de comunicação, o *publisher* (dispositivo que envia a mensagem) e o *subscriber* (dispositivo que recebe a mensagem), estão completamente desacoplados e toda troca de mensagens entre eles deve passar por um *broker* intermediário (TRUONG; L., 219; ANDY *et al.*, 2017; FIRDOUS *et al.*, 2018; LEKIĆ *et al.*, 2019).

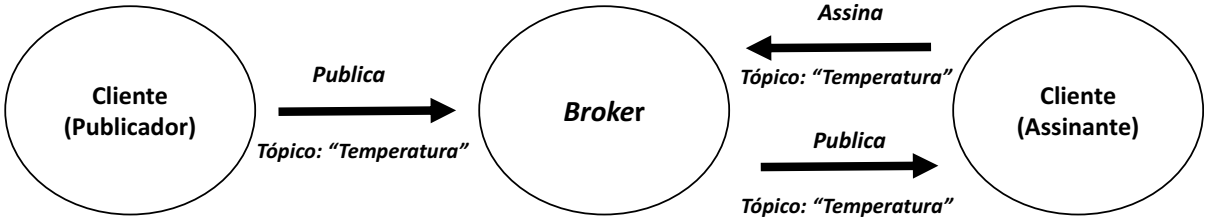
Este desacoplamento permite que o remetente não saiba quem é o receptor (IP, porta, localização), bem como não é necessário que eles estejam ativos ao mesmo tempo para estabelecer

[1] O HTTP (*Hypertext Transfer Protocol*) é um protocolo de camada de aplicação que opera na porta 80/TCP, usado em aplicações *web*, com a desvantagem de gerar uma sobrecarga quando usado em dispositivos limitados, como os de IoT (DIRO *et al.*, 2020).

uma troca de mensagens (DIRO *et al.*, 2020). A responsabilidade de manipular a comunicação entre dois pontos diferentes e desconhecidos é realizada pelo *broker* (FIRDOUS *et al.*, 2018; PERALTA *et al.*, 2017). A Figura 2 apresenta o paradigma Publicação e Assinatura presente na arquitetura do protocolo MQTT, em que um *publisher* (publicador) comunica-se com um *subscriber* (assinante) por meio de mensagens que trafegam através de um *broker*.

A comunicação neste paradigma se dá por meio de tópicos de mensagens. Tópicos são estruturas textuais hierárquicas (*strings*), separadas por "/", destinadas a organizar os processos de publicação e assinatura conduzidos pelo *broker* (NAZIR; KALEEM, 2019; TOLDINAS *et al.*, 2019). Os tópicos são criados pelos *publishers* no momento da publicação de uma mensagem e devem ser especificados pelos *subscribers* durante uma assinatura. Apenas os dispositivos inscritos em algum tópico específico podem se comunicar com o *broker*. Um dispositivo cliente é caracterizado por iniciar uma conexão com o *broker*, seja na assinatura ou na publicação de um tópico (TOLDINAS *et al.*, 2019; FIRDOUS *et al.*, 2018; LEKIĆ *et al.*, 2019).

Figura 2 – Arquitetura Publicação e Assinatura



Fonte: Adapdado de Peralta *et al.* (2017)

O protocolo MQTT utiliza o mínimo de largura de banda e lida com redes não confiáveis, exigindo pouco esforço de implementação por parte dos desenvolvedores (PALMIERI *et al.*, 2019). O MQTT atua em três níveis diferentes de Qualidade de Serviço ou *Quality Of Service* (QoS) , 0, 1 e 2, que em resumo, diferenciam-se pelo nível de confiabilidade na entrega das mensagens (POTRINO *et al.*, 2019; TRUONG; L., 219). Assinante e publicador podem definir um nível de QoS, e, caso exista divergência entre eles, o *broker* adota o QoS mais baixo. Os níveis de QoS e suas respectivas características são (TOLDINAS *et al.*, 2019):

- **QoS 1: *At most once* (No máximo uma vez)**: conhecido como *"fire and forget"* ("dispare e esqueça"). É o QoS padrão (*default*) do MQTT e é representado pelo número "0". O QoS 0 trabalha com a abordagem do mínimo esforço, pois não garante a entrega da mensagem, não gera sinal de *feedback* e não armazena a mensagem para um possível reenvio;

- **QoS 2: *At least once* (Ao menos uma vez)**: o nível de QoS 1 garante que uma mensagem seja entregue pelo menos uma vez ao destinatário. O remetente armazena a mensagem até receber um pacote PUBACK do destinatário, o que confirma o recebimento da mensagem. Neste QoS é possível que uma mensagem seja enviada ou entregue várias vezes, caso o reenvio ocorra antes da confirmação;

- **QoS 3: *Exactly once* (Exatamente uma vez)**: o nível de QoS 2 garante que a mensagem seja entregue exatamente uma única vez, ou seja, não ocorre duplicidade no recebimento de mensagens pelo receptor (*subscribe*). Para garantir a confiabilidade desse QoS é necessário o envio de 2 pares de confirmação de recebimento, conhecido como *four-part handshake*, o que ocorre nos dois sentidos e para todas as mensagens enviadas. Esta dualidade na confirmação entre emissor e receptor garante a exatidão do recebimento da mensagem uma única vez, não ocorrendo possíveis duplicações de recebimentos como no QoS 1. Enquanto o recebimento de uma mensagem não é confirmado pelo receptor, ela não é descartada no emissor.

Em níveis mais altos de QoS, o consumo de energia torna-se elevado, chegando a ter seu valor dobrado ao utilizar o QoS 2 em substituição ao QoS 0 (os dois extremos) (TOLDINAS *et al.*, 2019). Isto pode ser um problema, uma vez que os dispositivos de IoT lidam com baixo nível de energia, memória limitada e pouca capacidade de processamento (SHAPSOUGH *et al.*, 2018).

2.2.1 Modelo de Ameaças

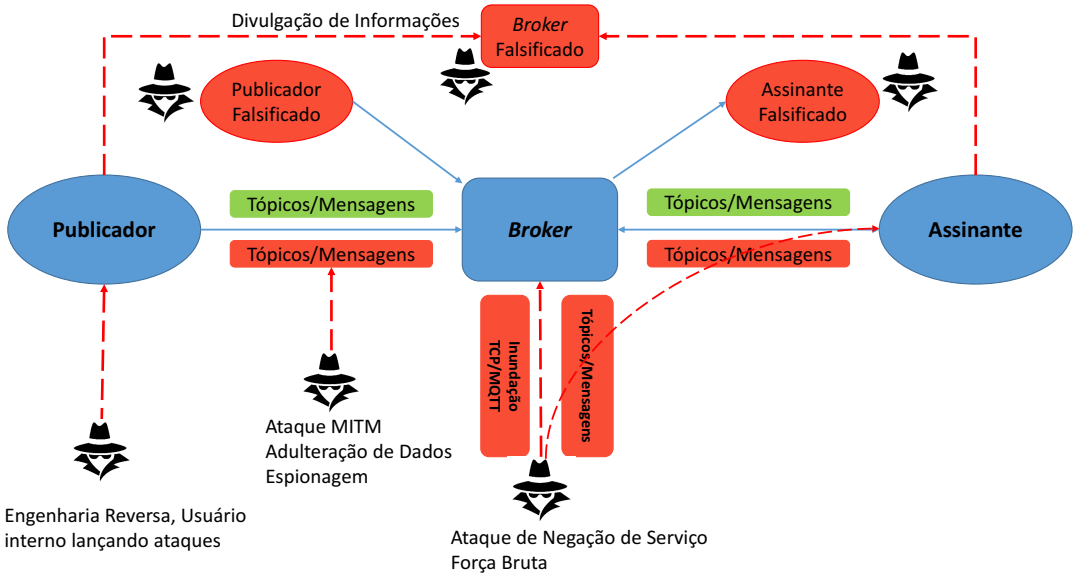
O protocolo MQTT suporta algumas camadas de segurança. Entretanto, a implementação destas camadas pode representar uma considerável sobrecarga nas comunicações (FIRDOUS *et al.*, 2018).

A compreensão das diferentes ameaças a que o MQTT está sujeito e o seu impacto na computação em névoa contribui para o desenvolvimento de medidas de detecção e mitigação de ataques, de acordo com o nível de dano em que ocorrem (FIRDOUS *et al.*, 2018). A grande oferta de sensores e atuadores em redes IoT atenua um possível ataque que tenha como alvo esses componentes de borda, causando pouco ou nenhum dano efetivo. Devido a isto, os atacantes centralizam esforços nos nós de névoa concentradores (POTRINO *et al.*, 2019; FIRDOUS *et al.*, 2018). Quando o equipamento vitimado por um ataque cibernético é o *broker*, a troca de

mensagens entre sensores e atuadores pode ficar completamente impossibilitada. A Figura 3 traz a representação do modelo de ameaças do protocolo MQTT na computação em névoa, onde as principais ações são direcionadas ao dispositivo central da névoa. Abaixo, estão elencados alguns tipos de ataques presentes no modelo de ameaças do MQTT na computação em névoa (POTRINO *et al.*, 2019; FIRDOUS *et al.*, 2018; Hernández Ramos *et al.*, 2018; ANDY *et al.*, 2017):

- Ataque de Negação de Serviço em nós de névoa;
- Danos físicos aos dispositivos de IoT;
- Exploração de mensagens com nível de QoS elevado;
- Interceptação de informações via ataque do Homem do Meio, do inglês *Man-in-the-Middle* (MIM);
- Adulteração de dados (*Data Tampering*);
- Engenharia reversa;
- Inserção de nós falsificados na rede (*Identity Spoofing*);

Figura 3 – Modelo de Ameaças na Computação em Névoa



Fonte: Adapdado de Firdous *et al.* (2018)

O ataque de negação de serviço, em todas as suas variações, é o tipo de ameaça que explora com mais frequência as vulnerabilidades MQTT (TIAN, 2020; BRUN *et al.*, 2019;

FIRDOUS *et al.*, 2018). Existem dois tipos de ataques de negação de serviço quando classificados a partir da origem dos ataques: ataques maliciosos que vêm de uma única fonte são chamados de ataques de negação de serviço ou *Denial of Service* (DoS) (HARIPRIYA; KULOTHUNGAN, 2019); os ataques que se originam de fontes múltiplas são conhecidos como ataques de negação de serviço distribuído (DDoS) (KEPCEOGLU *et al.*, 2019). Os ataques também são classificados de acordo com o impacto em recursos, largura de banda, infraestrutura de rede, entre outros, demonstrando uma grande variação de métodos e escopo de atuação (BRUN *et al.*, 2019; ROOHI *et al.*, 2019).

Normalmente, os ataques DoS/DDoS visam tornar indisponíveis os recursos da rede, promovendo atraso ou interrupção de um serviço para usuários válidos ou consumindo recursos de *hardware* de um alvo (SIKORA *et al.*, 2019; SHI *et al.*, 2019). Na Computação em Névoa, dada a importância de um *broker*, um ataque DDoS de natureza volumétrica pode interromper o tráfego completo de dados entre diferentes processos. Um ataque malicioso com este comportamento tende a consumir os recursos físicos deste dispositivo central (VISHWAKARMA; JAIN, 2020). Dentre todas as versões volumétricas de DDoS, as seguintes podem ser destacadas como as mais presentes em ambientes que utilizam o protocolo MQTT:

- Ataque de **Inundação por Pacotes SYN**: este é o tipo de ataque DDoS que, em 2018, se fez presente em mais de 80% dos registros (SANGODOYIN *et al.*, 2018; VISHWAKARMA; JAIN, 2020). A Inundação SYN TCP (TCP SYN *Flood*) é caracterizada por explorar as lacunas de segurança do *three-way handshake* do protocolo TCP (FIRDOUS *et al.*, 2018). Quando um *host* recebe uma solicitação de conexão mediante um pacote SYN, ele deve lidar com essa solicitação respondendo um SYN-ACK. O atacante explora esta falha, enviando várias solicitações SYN para a vítima, nunca respondendo com o ACK ao SYN-ACK enviado por ela (FIRDOUS *et al.*, 2018; BRUN *et al.*, 2019). A inundação propriamente dita ocorre quando o atacante envia pacotes a uma velocidade maior que o alvo pode tratar, preenchendo assim toda a sua tabela de conexões TCP, dificultando ou impedindo novas conexões legítimas (DULIK, 2019);

- Ataque de **Inundação por Pacotes Grandes**: como o MQTT permite uma carga útil (*payload*) de até 256 MB, usuários mal-intencionados podem enviar várias mensagens que excedam esse limite, com o objetivo de consumir os recursos do *broker* (HARSHA *et al.*, 2018; NAZIR; KALEEM, 2019). Neste ataque, a vítima é inundada por pacotes

fragmentados. Este tipo de ataque pode ser combinado com a utilização de níveis de QoS elevados nas mensagens, causando ainda mais danos ao *broker* (POTRINO *et al.*, 2019);

- Ataque de **Inundação por Pacotes ICMP**: o ICMP é um protocolo que geralmente é responsável por gerar mensagens de erro a uma origem em relação ao estado inacessível de um determinado destino. Em uma inundação ICMP (ICMP *Flood*) é explorado o comportamento utilizado por este protocolo (KEPCEOGLU *et al.*, 2019). O atacante envia diversos pacotes de solicitação, o *echo request*, mas não espera pela resposta do alvo, o *echo reply*. A inundação ocorre pela grande quantidade de pacotes enviados em um curto espaço de tempo (VISHWAKARMA; JAIN, 2020).

2.3 LÓGICA *FUZZY*

A teoria dos conjuntos *Fuzzy* destina-se a modelar incertezas e suas funções são aplicadas nas mais diferentes áreas. Na lógica *fuzzy*, tanto a subjetividade de um determinado dado quanto a experiência dos profissionais (especialistas) são consideradas. A lógica *fuzzy* opera com mapeamento não linear e lida com valores imprecisos, semelhantes à lógica humana (ZADEH, 1965; PEDRYCZ; GOMIDE, 1998; SHAH, 2018). A lógica *fuzzy* não é binária como a lógica clássica. A Equação 1 traduz uma das bases da teoria clássica dos conjuntos, onde o conceito de pertinência de um elemento *x* fica bem definido. Dado um conjunto *N* em um universo *X*, os elementos deste universo simplesmente pertencem ou não àquele conjunto (ZADEH, 1965).

$$f(x) = \begin{cases} 1 & \text{se, somente se, } x \in N \\ 0 & \text{se, somente se, } x \notin N \end{cases} \tag{1}$$

A lógica *fuzzy* permite infinitos valores intermediários compreendidos entre as duas extremidades do conceito de falso e verdadeiro, chamados de graus de pertinência. O grau de pertinência de um elemento *x*, dentro de um universo do discurso *X* em um conjunto *fuzzy N* é mapeado para um valor entre 0 e 1 e expressa o quanto este elemento pertence ao conjunto. O grau de pertinência pode ser representado por um conjunto de pares ordenados visto na Equação 2, onde μN(x) indica o quanto *x* é compatível com o conjunto *N* (NAIK, 2015; ZADEH, 1965).

$$N = \left\{ \mu_N(x)/x \right\} x \in X \tag{2}$$

A Equação 3 demonstra que um elemento *x* pode pertencer completamente, não perten-

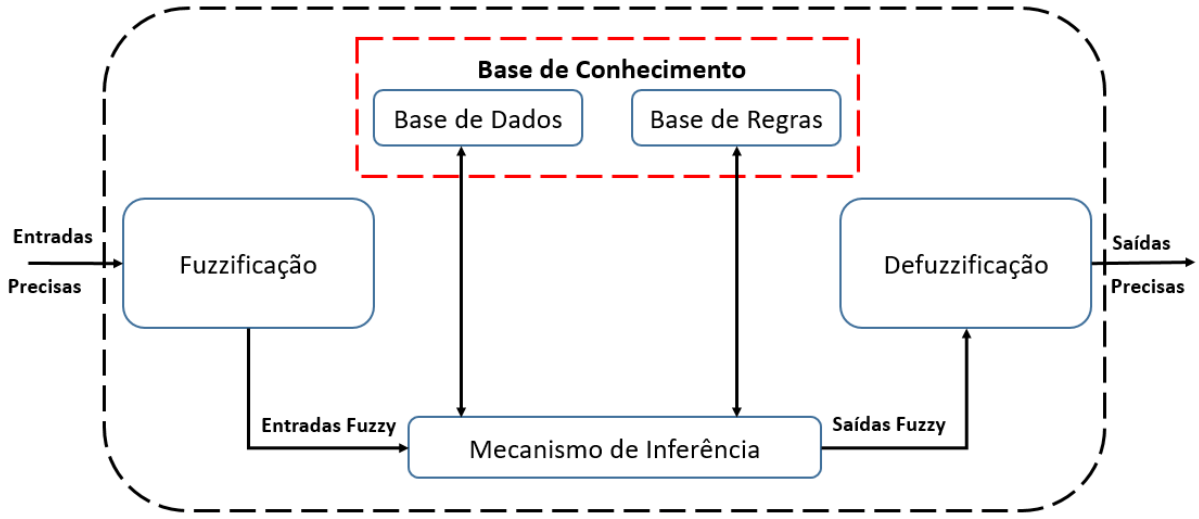
cer, ou ainda, pertencer parcialmente ao conjunto *N*.

$$f(x) = \begin{cases} 1, & \text{se, e somente se } x \in N \\ 0, & \text{se, e somente se } x \notin N \\ 0 \leq \mu(x) \leq 1, & \text{se } x \text{ pertence parcialmente a } N \end{cases} \tag{3}$$

Em um Sistema de Inferência *Fuzzy* (SIF), visto de um plano cartesiano, o eixo "x" representa o universo do discurso, enquanto o eixo "y" representa a pertinência no intervalo $[0, 1]$ (PEDRYCZ; GOMIDE, 1998).

Um SIF é compreendido por quatro unidades (ver Figura 4): Fuzzificação, Mecanismo de Inferência (Raciocínio), Base de Conhecimento e Defuzzificação (ZIMMERMANN, 2001; PEDRYCZ; GOMIDE, 1998; BOUGDIRA *et al.*, 2019).

Figura 4 – Sistema de Inferência *Fuzzy*.



Fonte: Adaptado de Berouine *et al.* (2019)

2.3.1 Unidade de Fuzzificação

Em uma aplicação padrão, as entradas de um SIF são valores não *fuzzy*, também chamados de valores *crisp*. Na unidade de fuzzificação, esses valores *singletons* de entrada são convertidos em conjuntos *fuzzy*, de acordo com suas funções de pertinência (SHAH, 2018; YOUSAF *et al.*, 2017; ZADEH, 1965). As funções de pertinência mais comuns são trapezoidal, gaussiana e triangular (LIAO *et al.*, 2009).

As variáveis linguísticas de entrada são criadas na unidade de fuzzificação. Variáveis

linguísticas são variáveis que assumem valores linguísticos, representados por funções de pertinência definidas no Universo de Discurso. O universo do discurso é definido como sendo os limites de valores máximos e mínimos que uma variável pode receber. Cada universo de discurso de uma variável linguística é composto por vários Termos Linguísticos. Um termo linguístico é uma expressão que dá nome às várias características que uma variável liguística pode assumir em um SIF (BOUGDIRA *et al.*, 2019; COSTA *et al.*, 2012; PEDRYCZ; GOMIDE, 1998; ZADEH, 1965). Essas características representam conceitos (por exemplo, "temperatura","peso", etc.), aceitam modificadores linguísticos (por exemplo, "ligeiramente", "muito", "pouco", etc.) e conectores lógicos (por exemplo "AND", "OR" e "NOT") (BOUGDIRA *et al.*, 2019; COSTA *et al.*, 2012; MAMDANI; ASSILIAN, 1975). Cada valor preciso de entrada deve ser comparado com as variáveis linguísticas presentes numa determinada regra *fuzzy* e criadas na unidade de fuzzificação para revelar o grau de compatibilidade entre eles (PEDRYCZ; GOMIDE, 1998; HARIPRIYA; KULOTHUNGAN, 2019). Esta comparação é chamada *Matching* dos Antecedentes.

2.3.2 Base de Conhecimento

A Base de Conhecimento é composta por duas estruturas fundamentais: a Base de Regras e a Base de Dados (COSTA *et al.*, 2012; ZIMMERMANN, 2001; PAPPIS; SIETTOS, 2005). A base de regras é formada pela junção de premissas definidas pelo especialista ou mediante análise de resultados experimentais. Essas regras são unidas por condicionais lógicos como "Se", "Se-Então", "Senão" e "Senão-Se" (*If, If-Then, Else, Else-If*) (SUGUNA *et al.*, 2018; SELVACHANDRAN *et al.*, 2019; ZADEH, 1965). A base de dados contém as definições numéricas necessárias para o estabelecimento das funções de pertinência usadas pelo conjunto de regras *fuzzy* (BOUGDIRA *et al.*, 2019; ZIMMERMANN, 2001).

2.3.3 Mecanismo de Inferência

O Mecanismo de Inferência dá origem aos conjuntos *fuzzy* destinados à fase de defuzzificação. A fase de Controle, como também é chamada esta etapa, é alimentada por parâmetros da base de conhecimentos e por conjuntos *fuzzy* oriundos da fuzzificação. Nesta etapa ocorre a agregação entre os parâmetros de entrada, oriundos da fuzzificação, chamada de agregação dos antecedentes (BOUGDIRA *et al.*, 2019; PEDRYCZ; GOMIDE, 1998; ZIMMERMANN, 2001). O resultado oriundo da agregação das variáveis antecedentes pode ser obtido a partir da

aplicação da função máximo (exemplo de norma-s) ou mínimo (exemplo de norma-t), de acordo com o operador utilizado, se “OU” ou “E”, respectivamente (ESPINOSA; VANDEWALLE, 2000; ZIMMERMANN, 2001; ALOMAR; ALAZZAM, 2019; PAPPIS; SIETTOS, 2005), como pode ser observado abaixo:

- **norma-t**: μA E μB = *min*(μA, μB)
- **norma-s**: μA OU μB = *max*(μA, μB)

Cada regra pode ser ativada através de uma semântica de ativação (LEE, 1990; SELVACHANDRAN *et al.*, 2019; ZIMMERMANN, 2001). As semânticas de Mamdani e Larsen são as duas semânticas de ativação existentes. A semântica da regra de ativação inferida pela conjunção Mamdani, dada pelo operador *min*, apresenta como saídas apenas as regras com um grau de ativação (nível de disparo da regra) maior do que zero (HARIPRIYA; KULOTHUNGAN, 2019; MAMDANI; ASSILIAN, 1975; PAPPIS; SIETTOS, 2005; SELVACHANDRAN *et al.*, 2019; SHAH, 2018). A semântica de Larsen, dada pelo operador produto, também apresenta saída para as regras com nível de disparo maior que zero, entretanto, o valor é menor quando comparado ao operador min. A ação final do Mecanismo de Inferência é a agregação das regras ativadas, também chamada de agregação das consequentes (PEDRYCZ; GOMIDE, 1998; SELVACHANDRAN *et al.*, 2019).

2.3.4 Unidade de defuzzificação

A unidade de defuzzificação pode fornecer como saída tanto um conjunto *fuzzy* como um valor numérico único. Qualquer uma dessas saídas é obtida a partir da agregação dos conjuntos *fuzzy* resultantes do mecanismo de inferência. Esta etapa de agregação produz um conjunto *fuzzy* que representa a saída *fuzzy* inferida pelo SIF. Nas aplicações que envolvem saída crisp, esse conjunto *fuzzy* deve ser defuzzzificado. Existem vários métodos de defuzzificação, como o centróide e a média dos máximos (BOUGDIRA *et al.*, 2019; ESPINOSA; VANDEWALLE, 2000; PAPPIS; SIETTOS, 2005; YOUSAF *et al.*, 2017; ZIMMERMANN, 2001).

2.4 TRABALHOS RELACIONADOS

Esta seção apresenta o estado da arte relacionado à segurança cibernética e ao gerenciamento de consumo de energia em ambiente IoT e na computação em névoa. Este último, com

foco nos trabalhos que propõem o uso da lógica *fuzzy*.

A comparação entre diferentes técnicas de ataques DDoS com o objetivo de avaliar a resiliência do sistema alvo é vista em Chen *et al.* (2018), Liang *et al.* (2017), Su *et al.* (2017) e Bao *et al.* (2016). A análise do consumo de recursos de *hardware* em dispositivos restritos é vista em Li *et al.* (2019) e Kepceoglu *et al.* (2019). A detecção e mitigação de ataques DDoS em *brokers* MQTT é contemplada em Sikora *et al.* (2019) e Palmieri *et al.* (2019). O estudo proposto por Palmieri *et al.* (2019) apresenta uma ferramenta que detecta automaticamente diversas vulnerabilidades em *brokers* MQTT e apresenta algumas alternativas de mitigação. A proposta baseia-se nas vulnerabilidades conhecidas do protocolo MQTT. São executados diversos ataques e os resultados são compilados em relatórios. Para cada vulnerabilidade explorada no *broker*, uma solução de mitigação foi proposta. Ataques DDoS, nós falsificados, adulteração de dados e ataque MITM foram algumas das vulnerabilidades exploradas. A proposta adotou algumas medidas de mitigação a estas vulnerabilidades, como a implementação de criptografia SSL/TLS, adoção de senhas e a limitação no tamanho de pacotes.

Mohammed *et al.* (2020) utiliza a lógica *fuzzy* no controle do consumo energético em um sistema de saúde portátil (*Healthcare System*) que atua em tempo real. A execução do sistema *fuzzy* se dá na tomada de decisões, selecionando quais tipos de dados de saúde serão enviados e a sua respectiva frequência de envio. Os dados de entrada são oriundos de sensores ligados ao corpo do paciente. Esses sensores captam alterações fisiológicas dos pacientes e remetem-nas ao sistema *fuzzy*. São aplicadas quatro regras *fuzzy*. O SIF proposto neste trabalho relacionado é capaz de reduzir em 30% o consumo energético médio diário do sistema, prolongando o tempo de assistência ao paciente.

O gerenciamento no consumo de energia elétrica proporcionado pela implementação da lógica *fuzzy* é proposto também por Chaouch *et al.* (2019). Sensores de temperatura e umidade fornecem as entradas necessárias para que um SIF gerencie um ar-condicionado inteligente. A proporção das grandezas de entrada do SIF extraídas do ambiente, temperatura e umidade, controlam a frequência da rotação do motor, elevando ou diminuindo os valores aplicados na saída. O sistema possui onze regras *fuzzy*, comunica-se em um ambiente ZigBee, utiliza o Raspberry Pi como *gateway* e o Arduino como microcontrolador de tensão. O experimento mostrou a eficácia da lógica *fuzzy* na economia de consumo energético, atuando no controle de temperatura em um ambiente e interagindo com os dispositivos de IoT.

Ciente da necessidade de economia de consumo energético em dispositivos restritos,

a proposta trazida por Shah (2018) apresenta um roteamento eficiente de energia, destinado a sistemas de IoT, que utiliza regras de inferência *fuzzy*. O objetivo desta proposta é aumentar a vida útil de uma rede IoT proporcionada pelo prolongamento de recursos energéticos em seus nós. O experimento contou com diversas métricas atuando como variáveis de entrada do SIF, tais como: velocidade de transmissão de dados, energia consumida e número de saltos para entrega de dados. A estrutura *fuzzy* foi composta por dezoito regras, adotaram-se funções de pertinência do tipo triangular, utilizou-se o Mamdani como semântica e o centróide como método de defuzzificação.

O trabalho proposto por Hernández Ramos *et al.* (2018) cria uma ferramenta que utiliza a lógica *fuzzy* para avaliar o comportamento de dispositivos inteligentes que utilizam o protocolo MQTT. O experimento considerou diferentes *brokers* servers, dentre eles o Mosquitto, o HiveMQ e o Mosca. Dentre as vulnerabilidades detectadas, o ataque DDoS foi o que gerou o maior impacto. O tráfego de rede e o consumo de CPU foram analisados. O trabalho mostrou que o Mosquitto foi o *broker* mais resiliente a ataques cibernéticos.

Haripriya e Kulothungan (2019) propõem um esquema de detecção de intrusão em computação em névoa baseado em lógica *fuzzy*. O estudo possui um SIF que contém duas variáveis antecedentes (baseadas na análise do tráfego de rede) e uma consequente (anomalia). O sistema proposto possui nove regras e avalia o desempenho do *broker* MQTT sob ataque DDoS. O modelo de análise adotado visa identificar possíveis ataques contra o *broker*, analisando o comportamento de mensagens de conexão MQTT (*CONNECT* e *CONNACK*).

Kepceoglu *et al.* (2019) analisam o consumo energético em dispositivos com recursos de *hardware* restritos durante ataques DDoS. Em uma *Local Area Network* (LAN) formada por quatro equipamentos de IoT conectados, foram executadas duas categorias de ataques DDoS, a Inundação por pacotes SYN e a Inundação por pacotes ICMP. A utilização de CPU e o tráfego de rede foram os parâmetros analisados.

O estudo de Shi *et al.* (2019) propõe a detecção de diferentes modalidades de ataques cibernéticos que produzem danos físicos, baseado na alteração do consumo energético em um Raspberry Pi. As informações são processadas em duas fases. A primeira fase tem a função de detectar anomalias no padrão de consumo. A segunda fase identifica o tipo de ataque baseado no nível de energia consumido. São emulados seis dos tipos mais comuns de ataques (vírus, intrusão, negação de serviço, aquecimento excessivo, cavalo de tróia e queda de energia). O tempo necessário para detecção de um ataque é de 180 segundos.

A comparação entre diferentes técnicas de ataques de negação de serviço é tratada por Sangodoyin *et al.* (2018). Um *Intrusion Detection System* (IDS) é criado para uma Rede Definida por Software, do inglês *Software Defined Networking* (SDN), a qual possui seus dispositivos atacados por três modos diferentes de DoS: inundação SYN, inundação ACK e a inundação HTTP. Um ataque é identificado mediante uma alteração significativa na latência de rede, obtida na comparação entre um ambiente sem ataque e um com ataque. O interessante a ser destacado neste estudo é que a identificação de ataques volumétricos realizados por longos períodos de tempo desfavorece a precisão da detecção, quando esta é realizada apenas mediante análise de tráfego de dados.

O modelo de ameaças ao protocolo MQTT é abordado por Firdous *et al.* (2018), onde são elencadas algumas de suas principais ameaças na computação em névoa, a qual, aponta o DDoS como a principal delas. Dada a importância de um *broker*, o estudo avalia a resiliência deste dispositivo central sob diferentes ataques DDoS: inundação SYN, inundação MQTT e inundação por Pacotes Grandes. As mudanças no comportamento de utilização da memória e CPU são analisadas para cada tipo de ataque.

Brun *et al.* (2019) implementam um tipo de IDS destinado a detectar diversos ataques, como DoS e privação de sono, em uma rede IoT. O experimento conta com um *gateway* IoT, que simula uma camada de névoa responsável por prover acesso à nuvem aos dispositivos de borda.

O trabalho proposto por Li *et al.* (2019) intitula-se como o primeiro a abordar detecção de ataques físicos e cibernéticos via auditoria de energia. A proposta utiliza várias unidades de Raspberry Pi, as quais foram expostas a ataques dos tipos físicos e cibernéticos. O ataque físico é realizado com o aquecimento excessivo oriundo de fonte externa, já o DDoS exemplifica uma modalidade de ataque cibernético. O sistema inicialmente aprende como se dá o consumo energético em situação normal e, posteriormente, sinaliza a ocorrência de algum tipo de alteração, caso ela ocorra. As métricas de entrada são definidas pelo consumo energético do processador, da interface de rede e do disco rígido. Os resultados foram categorizados em ataque físico, ataque cibernético e não ataque.

O Quadro 1 apresenta as principais características encontradas nos trabalhos relacionados. Os parâmetros de análise expressam de maneira resumida as métricas exploradas em cada trabalho e seus respectivos ambientes de atuação. Esse Quadro apresenta também uma perspectiva sobre segurança ao abordar a existência, ou não, de detecção de ataque e identificação do tipo desse ataque. A lógica *fuzzy* é utilizada em alguns desses estudos na condução de análise

energética, controle de temperatura e categorização de ataques.

Quadro 1 – Resumo das Principais Características dos Trabalhos Relacionados

Trabalhos Relacionados	Parâmetros Analisados	Ambiente de Atuação e Objetivo	Segurança	Adoção de *Fuzzy*
Palmieri *et al.* (2019)	Vulnerabilidades conhecidas do MQTT.	IoT/Névoa (*broker* MQTT). Detectar e mitigar vulnerabilidades.	Detecção e mitigação de ataques: DDoS, nós falsificados, adulteração de dados e MITM.	-
Mohamed *et al.* (2020)	Alterações fisiológicas dos pacientes.	IoT (sistema de saúde portátil). Controle de consumo energético.	-	Sim
Chaoch *et al.* (2019)	Temperatura e umidade de um ambiente.	IoT (ar-condicionado inteligente). Controle de consumo energético.	-	Sim
Shah *et al.* (2018)	Velocidade de transmissão de dados, energia consumida e número de saltos utilizados na entrega de dados.	IoT. Controle de consumo energético.	-	Sim
Hernandéz Ramos *et al.* (2018)	Tráfego de rede e utilização de CPU.	IoT/Névoa (*broker* MQTT). Detectar vulnerabilidades.	Detecção de ataques DDoS.	Sim
Haripriya e Kulothungan *et al.* (2019)	Tráfego de rede.	IoT/Névoa (*broker* MQTT). Detectar e identificar vulnerabilidades.	Detecção e identificação de ataques DDoS.	Sim
Kepceoglu *et al.* (2019)	Tráfego de rede e utilização de CPU.	IoT. Analisar o consumo energético em dispositivos sob ataques.	Detecção de ataques DDoS.	-
Shi *et al.* (2019)	Consumo energético de dispositivos.	IoT/Névoa. Detectar e identificar anomalias.	Detecção e identificação de anomalias: vírus, intrusão, DoS, aquecimento, cavalo de Troia, queda de energia.	-
Sangotoyin *et al.* (2018)	Latência de rede.	SDN. Detectar e identificar anomalias.	Detecção e identificação de ataques DoS.	-
Firdous *et al.* (2018)	Utilização de CPU e memória.	IoT/Névoa. Detectar e identificar ataques.	Detecção e identificação de ataques DDoS.	-
Brun *et al.* (2019)	Tráfego de rede.	IoT/Névoa. Implementar um IDS e detectar ataques.	Detecção de ataques DoS e de privação de sono.	-
Li *et al.* (2019)	Consumo energético do processador, interface de rede e do disco rígido.	IoT/Névoa. Detectar e identificar ataques.	Detecção e identificação de ataques: DDoS e aquecimento.	-

Fonte: Autoria própria.

O objetivo do presente trabalho é detectar e identificar diferentes tipos de ataques DDoS em um nó da névoa através de um sistema de inferência *Fuzzy* que recebe como entrada a média de consumo energético desse nó em relação ao tempo. O próximo capítulo apresenta a arquitetura proposta.

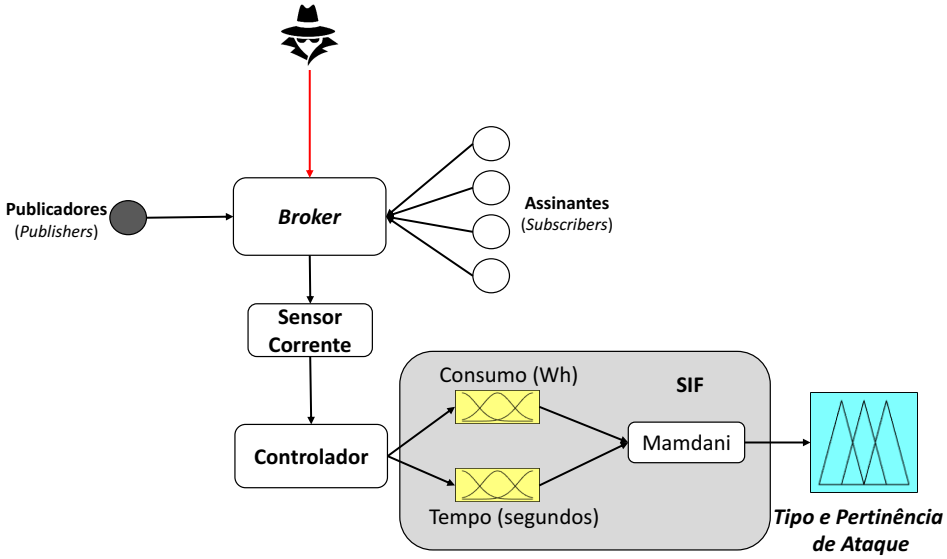
3 A ARQUITETURA PROPOSTA PARA COMPUTAÇÃO EM NÉVOA

O presente trabalho identifica três modalidades de ataques volumétricos de negação de serviço que visam drenar recursos em um *broker* MQTT. O objetivo é monitorar o consumo de energia da vítima, detectar e identificar um ataque cibernético com base nas mudanças no padrão desse consumo. Os diferentes tipos de ataques possuem comportamento energético característicos. A lógica *fuzzy* é utilizada na avaliação da letalidade de cada uma dessas ameaças, identificando-as e inferindo seus respectivos graus de pertinência, com base na alteração do padrão de consumo promovido por cada um desses ataques. A tomada de decisão e a adoção rápida de contramedidas para cada tipo de ataque podem ser melhoradas com o conteúdo deste estudo. O SIF proposto utiliza Mamdani como semântica de ativação e o centróide como método de defuzzificação.

Para medir o consumo de energia de um *broker* e validar o SIF utilizado para identificar ataques DDoS em um ambiente de computação em névoa, é proposta a arquitetura ilustrada na Figura 5. Esta arquitetura consiste nos seguintes elementos:

- Cinco clientes MQTT: um publicador e quatro assinantes;
- Um *broker* MQTT com QoS 0, representado pelo *broker* Mosquitto Server. O Mosquitto é um dos *brokers* de mensagens MQTT mais usados pois é de código aberto, leve e adequado para uso em dispositivos de IoT com recursos limitados (HWANG *et al.*, 2019). O dispositivo físico responsável por desempenhar a função de *broker* MQTT é o Raspberry Pi 3 Modelo B. O Raspberry foi escolhido por ser um computador de baixo custo, de placa única e amplamente utilizado em implementações de IoT usando MQTT, desempenhando principalmente a função de *broker* (LEKIĆ *et al.*, 2019; NAZIR; KALEEM, 2019). O SIF proposto é adequado apenas para este modelo específico de equipamento. No entanto, o sistema pode ser adaptado para outros modelos de Raspberry;
- Um sensor de corrente ACS712, o qual é considerado o sensor mais preciso de sua categoria (SURYA KARTIKA *et al.*, 2019; KHWANRIT *et al.*, 2018);
- Um microcontrolador Arduino Uno;
- Um SIF capaz de detectar e identificar um tipo de ataque DDoS em um *broker* MQTT.

Figura 5 – Arquitetura proposta



Fonte: Autoria própria.

As três modalidades de ataque DDoS (Inundações SYN, ICMP e Pacotes Grandes) foram geradas separadamente durante a troca de mensagens MQTT entre os clientes e o *broker*. Cada ataque foi gerado usando a ferramenta *hping3*, um utilitário nativo do sistema operacional Kali Linux. O *hping3* é amplamente utilizado, principalmente por possuir maior versatilidade na condução dos ataques, possibilitando, dentre outras coisas, a simulação de ataques distribuídos (parâmetro: *--rand-source*) sendo originado de um único dispositivo físico (JONES *et al.*, 2020).

Durante cada condição do *broker*, ataque e não ataque, informações de consumo são coletadas. Os dados de corrente elétrica lidos pelo sensor ao longo do tempo (em mA) com uma tensão sempre constante são enviados ao microcontrolador. O microcontrolador é responsável por fornecer o consumo médio de energia elétrica (em Wh) e o intervalo de tempo decorrido (em segundos) como valores de entrada do SIF.

Não são coletados dados energéticos durante o processo de inicialização do *broker* MQTT, pois, neste momento, ele apresenta um alto consumo de energia, chegando a 2,76 Wh, corroborado por um uso intenso de seus recursos de *hardware*. Este nível de consumo é compatível com uma situação de ataque de inundação. Com isso, deve-se desconsiderar, para fins de cálculos, o tempo necessário para a finalização completa do *boot* do sistema, o que, neste modelo de Raspberry Pi, dura cerca de 30 segundos (MURZAEVA *et al.*, 2019). Para a entrada no SIF, são considerados apenas os valores colhidos após os primeiros 60 segundos, tempo necessário, com boa margem de segurança, para o *boot* completo do sistema e a inicialização de

todos os serviços essencias, dentre eles o Mosquitto Server.

3.1 O SISTEMA DE INFERÊNCIA *FUZZY*

Tanto o consumo médio de energia do *broker* MQTT quanto o intervalo de tempo decorrido compõem as variáveis de entrada do SIF. O universo de discurso da variável "consumo" foi definido como estando entre 0 e 3 Wh, por corresponder aos limites nominais da operação do Raspberry (BEKAROO; SANTOKHEE, 2016; PAVELIC *et al.*, 2018). A variável "tempo" possui um universo de discurso entre 0 e 60 segundos, sendo considerado um tempo suficiente para detectar e identificar com precisão um ataque DDoS e seu respectivo grau de pertinência. A variável de saída do SIF consiste na identificação do tipo de ataque DDoS. O grau de pertinência figura como um subproduto do resultado do SIF.

Na unidade de fuzzificação, são criadas as variáveis linguísticas de entrada ou antecendentes (consumo e tempo) e a variável linguística de saída ou consequente (identificação do ataque). Cada variável linguística de entrada tem cinco termos linguísticos (Consumo: Consumo Baixo (CB), Consumo Médio (CM), Consumo Médio Alto (CEA), Consumo Alto (CA), Consumo Muito Alto (CMA); Tempo: Tempo Muito Curto (TMC), Tempo Curto (TC), Tempo Médio (TM), Tempo Longo (TL), Tempo Muito Longo (TML)), totalizando 25 regras, que serão armazenadas na Base de Regras do SIF, representada pela Tabela 1. A variável de saída é formada por quatro termos linguísticos (Identificação de Ataque: Não Ataque (NA), Inundação por Pacotes Grandes (IPG), innundação ICMP (ICMP), e inundação SYN (SYN)). Esta variável de saída possui um universo de discurso formado por um intervalo de 100 elementos. A saída do SIF representa não apenas a identificação de um ataque, mas também seu grau de pertinência.

Os conjuntos *fuzzy* que delimitam as variáveis linguísticas de tempo e consumo foram elaborados pelo especialista. As variáveis linguísticas de entrada são formadas, cada uma, por cinco termos linguísticos e são representadas graficamente por funções de pertinência triangulares (*trimf (x, y, z)*, sendo *x* e *z* parâmetros de mínima pertinência e *y* de máxima pertinência) utilizadas devido sua simplicidade e pouca demanda computacional. Os termos linguísticos estão definidos de acordo com os seguintes intervalos:

Termos Linguísticos de Consumo (Wh)

- Consumo Baixo (CB) = [0; 0; 1,9];
- Consumo Médio (CM) = [1,8; 1,9; 2];

- Consumo Médio Alto (CEA) = [1,9; 2; 2,1];
- Consumo Alto (CA) = [2; 2,1; 2,2];
- Consumo Muito Alto (CMA) = [2,1; 2,5; 3].

Termos Linguísticos de Tempo (segundos)

- Tempo Muito Curto (TMC) = [0; 0; 15];
- Tempo Curto (TC) = [0; 15; 30];
- Tempo Médio (TM) = [15; 30; 45];
- Tempo Longo (TL) = [30; 45; 60];
- Tempo Muito Longo (TML) = [45; 60; 60].

A variável linguística de saída (consequente) representa o comportamento de um *broker* ao final do período de coleta de dados. Ela é caracterizada pela identificação de um tipo de ataque DDoS ou constatação de funcionamento normal do *broker*. Os conjuntos *fuzzy* que delimitam a variável linguística de saída também foram elaborados pelo especialista. A variável de identificação de ataque, representada pelo eixo horizontal (x) de um plano cartesiano, é formada por quatro termos linguísticos e é dada pelas seguintes funções de pertinência triangulares:

Termos Linguísticos de Ataque

- Não Ataque (NA) = [0; 0; 30];
- Inundação por Pacotes Grandes (IPG) = [25; 40; 55];
- Inundação ICMP (ICMP) = [50; 70; 90];
- Inundação SYN (SYN) = [85; 99; 99].

O grau de pertinência, representado pelo eixo vertical (y) de um plano cartesiano, aparece como um subproduto da saída e representa a intensidade com que o ataque chega ao *broker*. Como observado na Equação 3, vista na Seção 2.3, cada um dos termos linguísticos de saída pode assumir valores entre 0 e 1 como o grau de pertinência (PEDRYCZ; GOMIDE, 1998):

- **Grau de Pertinência** (Não Ataque, Inundação de pacote grande, inundação ICMP ou inundação SYN) = [0; 1];

A base de regras que compõe o mecanismo de inferência é formada por 25 regras, elencadas de “R1” a “R25”:

- **R1**: se Consumo Baixo e Tempo Muito Curto, então Não Ataque;
- **R2**: se Consumo Médio e Tempo Muito Curto, então Não Ataque;
- ...
- **R24**: se Consumo Alto e Tempo Muito Longo, então Ataque de inundação SYN;
- **R25**: se Consumo Muito Alto e Tempo Muito Longo, então Ataque de Inundação SYN;

A Tabela 1 mostra a relação entre os termos antecedentes do SIF. O cruzamento (linha x coluna) entre esses termos compõe as 25 regras do SIF. Os termos de saída representam o resultado do cruzamento entre os termos antecedentes.

Tabela 1 – Base de Regras proposta para o Sistema de Inferência *Fuzzy*

Variáveis Linguísticas (Antecedentes)	**TMC**	**TC**	**TM**	**TL**	**TML**
CB	NA	NA	NA	NA	NA
CM	NA	NA	NA	NA	NA
CEA	NA	NA	IPG	ICMP	SYN
CA	NA	NA	IPG	ICMP	SYN
CMA	NA	NA	IPG	ICMP	SYN

Fonte: Autoria própria.

A agregação dos antecedentes das regras é realizada pelo operador lógico “E” (norma-t), obtendo como resultado o valor mínimo entre os antecedentes (*min*).

A inferência da ativação da regra é dada pela semântica de Mamdani, pois apresenta como saídas apenas regras com grau de ativação maior que zero. Uma segunda etapa de agregação é realizada apenas com os consequentes, ou seja, as regras ativadas por Mamdani, usando o operador “OU” (norma-s), obtendo como resultado o valor máximo entre os consequentes (*max*).

O modelo de defuzzificação utilizado neste estudo é o centróide, em que o valor numérico obtido representa o centro de gravidade (ou massa) da distribuição do conjunto de saída *fuzzy* (YOUSAF *et al.*, 2017). Os rótulos que identificam as possíveis condições do *broker* (NA, IG, ICMP, SYN) foram ordenados no eixo “x” do plano cartesiano de forma crescente em relação ao consumo de energia, justificando assim a utilização do centróide como classificador *fuzzy*.

3.2 PARÂMETROS DE ATAQUES

As três modalidades de ataque DDoS selecionadas são geradas separadamente, tendo como vítima o *broker* MQTT. O tempo necessário para o SIF proposto detectar, identificar e inferir o grau de pertinência de um ataque DDoS é de 30 segundos. Entretanto, quanto maior for o tempo de coleta de dados mais claros serão os resultados obtidos, logo, valores entre 45 e 60 segundos representam uma melhor amostra de tempo. Os três tipos de ataques foram realizados a partir de um terminal Linux e seguiram as sintaxes:

Ataque de Inundação por Pacotes Grandes: *hping3 -d 300000 -p 1883 --flood 192.168.15.5*.

- *hping3*: comando;
- -d 300000: tamanho do pacote em bytes;
- -p 1883: porta de ataque;
- --*flood*: inundação de pacotes;
- 192.168.15.5: endereço do alvo;

Ataque de inundação de ICMP: *hping3 -1 192.168.15.5 --flood --rand-source*.

- *hping3*: comando;
- -1: indicação de modo ICMP;
- --*flood*: inundação de pacotes;
- --*rand-source*: endereços aleatórios de origem;
- 192.168.15.5: endereço do alvo;

Ataque de Inundação SYN: *hping3 -S -p 1883 --flood --rand-source 192.168.15.5*.

- *hping3*: comando;
- -S: envio da *flag* SYN;
- -p 20883: porta de ataque;
- --*flood*: inundação de pacotes;

- *--rand-source*: endereços aleatórios de origem;
- 192.168.15.5: endereço do alvo;

Este capítulo apresentou os elementos utilizados na arquitetura proposta para analisar o consumo de energia de um *broker* durante seu funcionamento normal e sob tipos diferentes de ataques DDoS. O SIF proposto foi validado a partir desta análise de modo a mostrar que este é capaz de detectar e identificar o tipo de ataque que incide no *broker* e inferir o seu grau de pertinência. Esses resultados são apresentados no próximo capítulo.

4 RESULTADOS

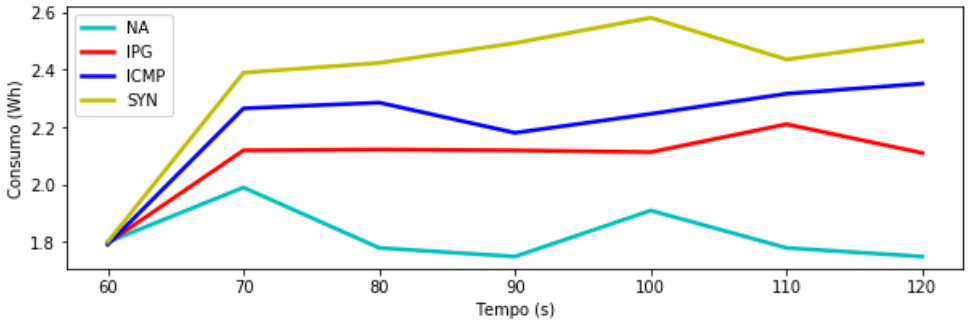
Este capítulo apresenta a análise do consumo de energia de um *broker* em operação normal e sob ataque e os resultados obtidos pelo SIF proposto para identificar um ataque DDoS e inferir seu respectivo grau de pertinência.

4.1 ANÁLISE DO CONSUMO DE ENERGIA DE UM *BROKER* MQTT

O consumo de energia de um *broker* MQTT, durante seu funcionamento normal e sob diferentes tipos de ataques DDoS, apresenta um comportamento bem característico, sendo possível distinguir entre cada um deles, como pode ser observado na Figura 6. Uma amostra de 60 segundos foi coletada, onde apresenta-se como resultado a comparação entre o consumo de energia do *broker* em operação normal (sem ataque) e quando submetido a cada um dos três tipos de ataques DDoS, desferidos a partir dos 60 segundos.

Os resultados foram obtidos a partir de 10 execuções de cada uma das condições impostas ao *broker* (NA, IPG, ICMP, SYN). Cada ponto "t" no gráfico (intervalo a cada 10 segundos) representa a média aritmética simples entre essas 10 execuções, com os mesmos parâmetros de ataque.

Figura 6 – Consumo energético do *broker*



Fonte: Autoria própria.

A identificação de um ataque está diretamente relacionada à energia consumida pelo *broker*. Portanto, por meio da demanda energética durante cada ataque, o SIF é capaz de identificar o tipo de ataque DDoS e seu grau de pertinência. A Figura 6 exibe o comportamento energético do *broker* MQTT em todas as condições às quais é submetido. A inundação com pacotes grandes (IPG) provou ser o tipo de ataque mais brando, representando um aumento no consumo de energia do *broker* de 8,02% após 60 segundos. O consumo médio de energia do

broker em operação normal foi $1{,}87\ Wh \pm 0{,}08$ e o ataque de inundação de pacotes grandes consumiu, em média, $2{,}02\ Wh \pm 0{,}12$. A inundação por pacotes ICMP apresenta nível de impacto intermediário, quando comparada aos outros dois ataques DDoS. Em 60 segundos de inundação, a inundação por pacotes ICMP elevou o consumo do *broker* em 19,3%, apresentando uma média de $2{,}13\ Wh \pm 0{,}09$ ao longo deste período. Durante o ataque de inundação de pacotes SYN, o consumo médio de energia do *broker* foi de $2{,}45\ Wh \pm 0{,}04$. Este tipo de ataque apresentou a maior diferença (31,01%) no consumo de energia entre as situações de "ataque" e "sem ataque".

A Tabela 2 apresenta a média de consumo de energia do *broker* em cada situação: sem ataque e sob ataques de inundação por Pacotes Grandes, ICMP e SYN.

Tabela 2 – Consumo de Energia em cada Situação de Ataque

Ataque	**Consumo de Energia (Wh)**
Sem Ataque	1,87±0,08
Inundação por Pacotes Grandes	2,02±0,12
Inundação por Pacotes ICMP	2,13±0,09
Inundação por Pacotes SYN	2,45±0,04

Fonte: Autoria própria.

4.2 IDENTIFICAÇÃO DE UM ATAQUE DDOS E SEU GRAU DE PERTINÊNCIA

Para mostrar que o SIF proposto é capaz de detectar um ataque DDoS, identificar seu tipo e inferir seu grau de pertinência em um *broker* MQTT, consideramos um cenário hipotético onde este *broker* registra, durante 56 segundos, um consumo médio de energia de 2,1 Wh.

Considerando esses dados de tempo (56 segundos) e consumo de energia (2,1 Wh), o Algoritmo 1 apresenta o pseudocódigo da unidade de fuzzificação. As linhas 2 e 3 apresentam os valores *crisp* de entrada: consumo médio de energia (*consumo_entrada*) e tempo (*tempo_entrada*), respectivamente. O Universo de Discurso das variáveis de consumo (*ud_consumo*), tempo (*ud_tempo*) e ataque (*ud_tiposAtaques*) são criados nas linhas 6, 7 e 8, respectivamente .

Algoritmo 1 – Pseudocódigo da unidade de Fuzzificação: valores *crisp* de entrada e universo do discurso.

```
1: #Valores Crisp de Entrada
2: consumo_entrada = 2,1;
3: tempo_entrada = 56;
4:
5: #Universo de Discurso
6: ud_consumo = [0; 3];
7: ud_tempo = [0; 60];
8: ud_tiposAtaques = [0; 99];
```

Fonte: Autoria própria.

O Algoritmo 2 mostra a criação das variáveis linguísticas de entrada: consumo (*consumoBaixo, consumoMedio, consumoMedioAlto, consumoAlto* e *consumoMuitoAlto*) nas linhas 2 a 6; tempo (*tempoMuitoCurto*, *tempoCurto*, *tempoMedio*, *tempoLongo* e *tempoMuitoLongo*) nas linhas 9 a 13; e ataque (*naoAtaque, inundpkgrd, inundicmp* e *inundsyn*) nas linhas 16 a 19.

Algoritmo 2 – Pseudocódigo da unidade de Fuzzificação: criação das váriáveis linguísticas de entrada.

```
 1: #Variáveis Linguísticas de Consumo;
 2: consumoBaixo = [0; 0; 1,9];
 3: consumoMedio = [1,8; 1,9; 2];
 4: consumoMedioAlto = [1,9; 2; 2,1];
 5: consumoAlto = [2; 2,1; 2,2];
 6: consumoMuitoAlto = [2,1; 2,5; 3];
 7:
 8: #Variáveis Linguísticas de Tempo;
 9: tempoMuitoCurto = [0; 0; 15];
10: tempoCurto = [0; 15; 30];
11: tempoMedio = [15; 30; 45];
12: tempoLongo = [30; 45; 60];
13: tempoMuitoLongo = [45; 60; 60];
14:
15: #Variáveis Linguísticas de Ataque;
16: naoAtaque = [0; 0; 30];
17: inundpkgrd = [25; 40; 55];
18: inundicmp = [50; 70; 90];
19: inundsyn = [85; 99; 99];
```

Fonte: Autoria própria.

O Algoritmo 3 apresenta o *matching* dos antecedentes. O objetivo desta etapa é obter o grau de compatibilidade entre os valores *crisp* (exatos) de entrada e seus correspondentes universos de discurso. O *Matching* entre as variáveis antecedentes é dado pela função Python *interp_membership (x, y, z)* que compõe a biblioteca Python *Scikit-fuzzy* (SCIKIT-FUZZY, 2020). Esta função infere o nível de pertinência entre os parâmetros de entrada, que são: (x) o conjunto *fuzzy* que representa o universo do discurso; (y) o conjunto *fuzzy* que representa as variáveis linguísticas; e (z) o valor *crisp* de entrada. Para ter algum grau de correspondência, o valor *crisp* de entrada deve estar contido nos limites do universo de discurso das variáveis linguísticas relacionadas. No Algoritmo 3 estão presentes apenas as operações de *matching* entre as variáveis antecedentes que resultam em valores maiores que zero. O valor resultante da linha 3 indica um grau máximo de compatibilidade (1,0) entre os antecedentes *consumoAlto* e o valor de entrada para média de consumo (*consumo_entrada*). No entanto, este mesmo valor de entrada para a média de consumo (2,1 Wh) possui grau zero de combinação com os outros antecedentes de consumo: *consumoBaixo*, *consumoMedio*, *consumoMedioAlto* e *consumoMuitoAlto*. Este processo de combinação também é feito com a variável de entrada de tempo (56 segundos), onde

apenas os antecedentes *tempoLongo* (linha 8) e *tempoMuitoLongo* (linha 11) possuem grau de combinação superior a zero com esta variável de entrada, apresentando um valor resultante de 0,26 e 0,73, respectivamente. As outras variáveis antecedentes de tempo (*tempoMuitoCurto*, *tempoCurto* e *tempoMedio*) resultam em uma combinação de grau zero quando comparadas ao valor de entrada de tempo.

Algoritmo 3 – Pseudocódigo da unidade de Fuzzificação: *matching* dos antecedentes.

```
1:  #Matching das Variáveis de Consumo Antecedentes
2:  ...
3:  CA = interp_membership (ud_consumo; consumoAlto; consumo_entrada);
4:  #CA = interp_membership ([0; 3]; [2; 2,2]; 2,1);
5:  ...
6:  ...
7:  #Matching das Variáveis de Tempo Antecedentes;
8:  TL = interp_membership (ud_tempo; tempoLongo; tempo_entrada);
9:  # TL = interp_membership ([0; 60]; [30; 60]; 56);
10: ...
11: TML = interp_membership (ud_tempo; tempoMuitoLongo; tempo_entrada);
12: #TML = interpmembership ([0, 60], [45, 60], 56);
```

Fonte: Autoria própria.

O Algoritmo 4 apresenta o pseudocódigo da agregação dos antecedentes realizada pelo mecanismo de inferência do SIF. Esta agregação é realizada entre os valores resultantes do *Matching*, unidos pelo operador lógico "E" (norma-t). Como resultado, obtém-se o valor mínimo entre os antecedentes agregados. A agregação dos antecedentes resulta em 25 níveis de disparo de cada uma das regras que compõem a base de regras do SIF. O Algoritmo 3 mostrou que apenas o antecedente de consumo de energia *consumoAlto* e os antecedentes de tempo *tempoLongo* e *tempoMuitoLongo* foram afetados pelos valores *singletons* de entrada, resultando em graus de compatibilidade diferentes de zero (1,0, 0,26 e 0,73, respectivamente). As linhas 1 e 4 do Algoritmo 4 mostram os resultados das duas únicas agregações de antecedentes, entre as 25 geradas, que resultam em valores diferentes de zero (0,26 e 0,73, respectivamente).

Algoritmo 4 – Pseudocódigo do mecanismo de inferência: agregação dos antecedentes

```
1: TL_CA = min (TL; CA);
2: #TL_CA = min (0,26; 1,0);
3:
4: TML_CA = min (TML, CA);
5: #TML_CA = min (0,73; 1,0);
```

Fonte: Autoria própria.

As variáveis antecedentes, após agregadas, conforme mostrado no Algoritmo 4, dão origem aos níveis de disparo das 25 regras do SIF proposto. Estas regras serão ativadas conforme o nível de disparo definido pelo operador min (escolhido para agregar os antecedentes). A

semântica escolhida para o trabalho foi a de Mamdani (operador de conjunção = min) que define a saída individual inferida pela regra (o operador min define o corte do conjunto do consequente). O Algoritmo 5 apresenta o pseudocódigo referente à conclusão (inferência) de cada regra obtida pela semântica de Mamdani. Todas as 25 regras são expostas a esse mesmo método de inferência.

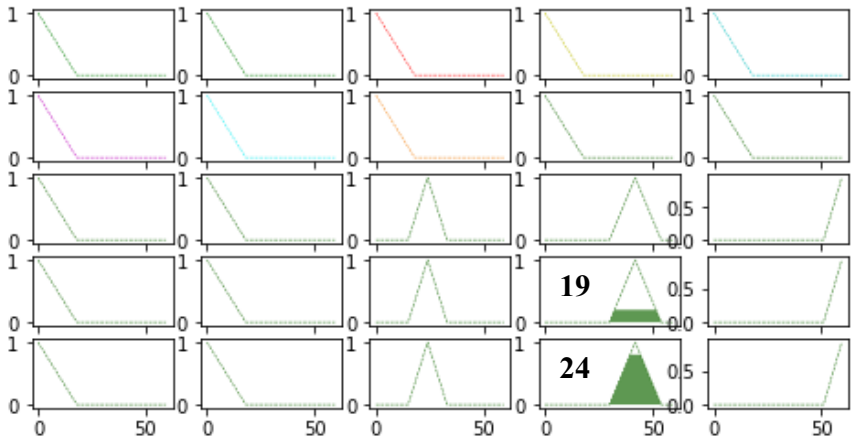
Algoritmo 5 – Pseudocódigo do mecanismo de inferência: ativação das regras

```
1: R19 = min (TL_CA; inundicmp);
2: #R19 = min (0,26, [0; 1])
3: ...
4: R24 = min (TML_CA; inundicmp);
5: #R24 = min (0,73; [0; 1])
6: ...
7: #Agregação das regras ativas por máximo;
8: regrasAtivas = max (R19; R24);
```

Fonte: Autoria própria.

Apenas as regras “R19” e “R24” são ativadas conforme os parâmetros de entrada adotados. A linha 1 do Algoritmo 5 mostra a semântica de Mamdani agindo na regra R19 e obtendo como resultado um conjunto *fuzzy* com valores entre 0 e 0,26. A linha 4 mostra a ação na regra R24, a qual resulta em um conjunto *fuzzy* com valores entre 0 e 0,73. A linha 8 mostra o término do mecanismo de inferência, caracterizado pela etapa de agregação dos consequentes. Nesta etapa o conjunto *fuzzy regrasAtivas* representa o resultado da união de cada uma das regras que foram ativadas por Mamdani e é obtido pelo operador *max* (norma-s), resultando no valor de 0,73. A representação gráfica da ativação das regras observadas nas linhas 1 e 4 do Algoritmo 5 pode ser vista na Figura 7.

Figura 7 – Ativação das Regras por Mamdani



Fonte: Autoria própria.

O conjunto *fuzzy* resultante da agregação das regras ativas (linha 8 do Algoritmo 5) atua como parâmetro de entrada da unidade de defuzzificação e determinará o grau de pertinência do

ataque.

O Algoritmo 6 mostra o pseudocódigo da unidade de defuzzificação. A saída do SIF proposto, obtida através da defuzzificação, é composta por dois resultados numéricos precisos. O primeiro identifica o tipo de ataque e o segundo infere o grau de pertinência em que esse ataque afeta o *broker*. A identificação do ataque (*id_ataque*), dada pela função *defuzz (x, y, z)* da biblioteca *Scikit-fuzzy* Python, é mostrada na linha 2 do Algoritmo 6. Esta função recebe os seguintes parâmetros de entrada: (x) o universo de discurso de ataques (*ud_tiposAtaques*), representando todos os tipos de ataques possíveis mapeados pelo SIF; (y) o conjunto resultante da agregação das regras ativas (*regrasAtivas*); e (z) o modelo de defuzificação usado no estudo (ou seja, o modelo de centróide). O conjunto *fuzzy* referente ao universo de discurso de ataques (*ud_tiposAtaques*) é um intervalo de 100 elementos (0-99), composto por quatro conjuntos (*naoAtaque, inundpkgrd, inundicmp* e *inundsyn*), onde a condição de cada tipo de ataque DDoS é mapeada, como descrito nas linhas 16-19 do Algoritmo 2. Portanto, através dos valores de entrada *crisp* (consumo de 2,1 Wh e tempo de 56 segundos), a função na linha 2 do Algoritmo 6 obteve o valor resultante de 71,14 como identificação de ataque. Este resultado significa uma inundação de pacote ICMP, uma vez que este valor está inserido no conjunto "*inundicmp = [50, 70, 90]*" (ver linha 18 do Algoritmo 2). A linha 6 do Algoritmo 6 mostra o grau de pertinência (*grau_pertinencia*) do ataque identificado. Essa associação é calculada pela função *interp_membership (x, y, z)* e recebe os seguintes parâmetros de entrada: (x) o universo de discurso de ataques (*ud_tiposAtaques*); (y) o conjunto resultante da agregação de regras ativas (*regrasAtivas*); e (z) o tipo de ataque previamente identificado (*id_ataque*). Considerando um *broker* com um consumo médio de energia de 2,1 Wh, medidos por 56 segundos, o SIF proposto indica um ataque de inundação de pacotes ICMP (*id_ataque* = 71,14) com um grau de pertinência de 0,73 (*grau_pertinencia* = 0,73).

Algoritmo 6 – Pseudocódigo da unidade de defuzzificação: identificação e pertinência do ataque

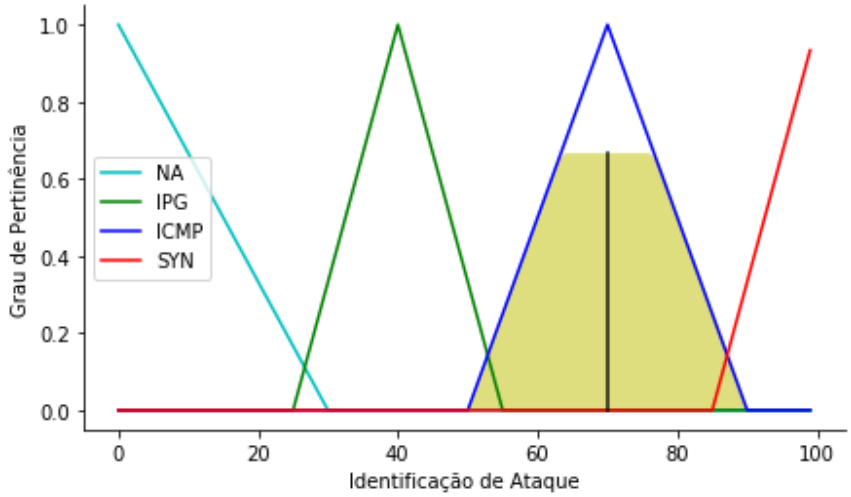
```
1: #Identificação do tipo de ataque;
2: id_ataque = defuzz (ud_tiposAtaques; regrasAtivas; centroide);
3: #id_ataque = defuzz ([0; 99]; [0; 0,73], centroide);
4: ...
5: #Pertinência do ataque;
6: grau_pertinencia = interp_membership (ud_tiposAtaques; regrasAtivas; id_ataque);
7: #gra_pertinência = Membership ([0; 99]; [0; 0,73]; 71,14);
```

Fonte: Autoria própria.

A Figura 8 mostra a saída do SIF onde a identificação do tipo de ataque está disposta no eixo horizontal e o grau de pertinência no eixo vertical. Como pode ser observado, o SIF

proposto indica um ataque de inundação de pacotes ICMP com um grau de pertinência de 0,73.

Figura 8 – Saída do Sistema de Inferência *Fuzzy*



Fonte: Autoria própria.

Nesse capítulo foi visto que o consumo de energia de um *broker* MQTT, durante seu funcionamento normal e sob diferentes tipos de ataques DDoS, apresenta comportamentos bem distintos. Portanto, considerando o consumo energético de um *broker* e o tempo desse consumo, o SIF proposto foi capaz de detectar um ataque DDoS, identificar o seu tipo e inferir o seu grau de pertinência.

5 CONCLUSÃO E TRABALHOS FUTUROS

O ataque DDoS é o mais recorrente e prejudicial aos dispositivos da Computação em Névoa, especialmente em nós de névoa centrais, como os *brokers*. Os *brokers* têm funções essenciais na camada de névoa, uma vez que eles provêm a comunicação entre os dispositivos IoT e a nuvem. Um ataque DDoS em um *broker* visa drenar recursos desse nó, como por exemplo sua energia, podendo representar atraso nas comunicações, interrupção temporária ou mesmo a indisponibilidade total do seu serviço afetando amplamente os dispositivos a ele conectados.

O objetivo deste trabalho é propor um Sistema de Inferência *Fuzzy* para detectar e identificar, com base nas mudanças no padrão de consumo de energia de um alvo, a modalidade de um ataque DDoS e inferir o grau de pertinência em que esse ataque atinge o alvo. A lógica *fuzzy* foi utilizada por lidar bem com valores imprecisos, quando comparada à lógica binária convencional, dispensando a adoção de cálculos matemáticos complexos.

Foram considerados três tipos de ataques DDoS: inundação de pacotes SYN, inundação de pacotes grandes e inundação de pacotes ICMP. Esses ataques volumétricos caracterizam-se por desgastar o alvo, forçando-o a despender um consumo energético anômalo, drenando seus recursos e depreciando seu desempenho. Nos experimentos realizados, esses ataques DDoS mostraram diferentes padrões de consumo de energia, o que permitiu que o SIF proposto detectasse e identificasse com precisão o tipo de ataque DDoS em um *broker* que trabalha com o protocolo MQTT. A inundação por pacotes SYN foi o ataque que gerou o maior consumo de energia. Este ataque consumiu 31,01% a mais de energia em comparação ao consumo do *broker* em uma operação normal (ou seja, sem ataques), seguido pelo ataque de inundação de pacotes ICMP com um aumento de 19,3% e o ataque de inundação de pacotes grande com um aumento de 8,02%.

No cenário proposto, com um consumo médio de energia de 2,1 Wh durante 56 segundos, o SIF foi capaz de detectar e identificar o tipo de ataque cibernético que atingiu o *broker*. Este cenário obteve como saída o valor de 71,14, o que representa um ataque de inundação de pacotes ICMP, de acordo com as regras *fuzzy* estabelecidas, e apresentou um grau de pertinência de 0,73.

Novas ameaças continuarão sendo desenvolvidas e as atuais sendo aprimoradas. Diante deste cenário, soluções de detecção, identificação e mitigação devem ser fomentadas. Como trabalho futuro propõe-se analisar o consumo energético dos nós de névoa sofrendo outros tipos de ataques cibernéticos e construir um SIF para identificar esses novos ataques e inferir seus

graus de pertinência. Também propõe-se analisar o comportamento de outros protocolos voltados para IoT e computação em névoa, como o CoAP, durante ataques cibernéticos.

REFERÊNCIAS

ABEDI, Masoud; POURKIANI, Mohammadreza. Resource Allocation in Combined Fog-Cloud Scenarios by Using Artificial Intelligence. **2020 5th International Conference on Fog and Mobile Edge Computing, FMEC 2020**, p. 218–222, 2020.

ACETO, Giuseppe; PERSICO, Valerio; PESCAPE, Antonio. A survey on Information and Communication Technologies for Industry 4.0: state of the art, taxonomies, perspectives, and challenges. **IEEE Communications Surveys & Tutorials**, IEEE, v. 1-, n. November 2011, p. 1–1, 2019.

ALABA, Fadele Ayotunde; OTHMAN, Mazliza; HASHEM, Ibrahim Abaker Targio; ALOTAIBI, Faiz. Internet of Things security: A survey. **Journal of Network and Computer Applications**, v. 88, n. December 2016, p. 10–28, 2017. ISSN 10958592.

ALOMAR, Ban; ALAZZAM, Azmi. A Smart Irrigation System Using IoT and Fuzzy Logic Controller. **ITT 2018 - Information Technology Trends: Emerging Technologies for Artificial Intelligence**, IEEE, p. 175–179, 2019.

ANDY, Syaiful; RAHARDJO, Budi; HANINDHITO, Bagus. Attack scenarios and security analysis of mqtt communication protocol in iot system. **International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)**, v. 4, n. September, p. 600–604, 2017. ISSN 2407439X.

ATLAM, Hany; WALTERS, Robert; WILLS, Gary. Fog Computing and the Internet of Things: A Review. **Big Data and Cognitive Computing**, v. 2, n. 2, p. 10, 2018. ISSN 2504-2289.

BAO, Cong; GUAN, Xingren; SHENG, Qiankun; ZHENG, Kai; HUANG, Xin. A Tool for Denial of Service Attack Testing in IoT. **st Conference on emerging topics in interactive systems, 2016.**, p. 5, 2016.

BEKAROO, Girish; SANTOKHEE, Aditya. Power consumption of the Raspberry Pi: A comparative analysis. **2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies, EmergiTech 2016**, IEEE, p. 361–366, 2016.

BEROUINE, A.; AKSSAS, E.; NAITMALEK, Y.; LACHHAB, F.; BAKHOUYA, M.; OULADSINE, R.; ESSAAIDI, M. A Fuzzy Logic-Based Approach for HVAC Systems Control. **2019 6th International Conference on Control, Decision and Information Technologies (CoDIT'19) | Paris, France / April 23-26, 2019**, IEEE, v. 1-, p. 1510–1515, 2019.

BONOMI, Flavio; MILITO, Rodolfo; ZHU, Jiang; ADDEPALLI, Sateesh. Fog computing and its role in the internet of things. **MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop**, p. 13–15, 2012.

BOUGDIRA, Abdesselam; AKHARRAZ, Ismail; AHAITOUF, Abdelaziz. Fuzzy approach to enhance quality control within intelligent traceability systems. **2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS 2019**, IEEE, 2019.

BRUN, Olivier; YIN, Yonghua; AUGUSTO-GONZALEZ, Javier; RAMOS, Manuel; BRUN, Olivier; YIN, Yonghua; AUGUSTO-GONZALEZ, Javier; RAMOS, Manuel; GELENBE, Erol; ATTACK, Iot; BRUN, Olivier; YIN, Yonghua; AUGUSTO-GONZALEZ, Javier; RAMOS, Manuel. IoT Attack Detection with Deep Learning To cite this version : HAL Id : hal-02062091 IoT Attack Detection with Deep Learning. **ISCIS Security Workshop, Feb 2018, Londres, United Kingdom**, 2019.

BUTUN, Ismail; OSTERBERG, Patrik; SONG, Houbing. Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. **IEEE Communications Surveys and Tutorials**, IEEE, v. 22, n. 1, p. 616–644, 2020. ISSN 1553877X.

CHAOUCH, Haithem; BAYRAKTAR, Abdullah Salih; ÇEKEN, Celal. Energy Management in Smart Buildings by Using M2M Communication. **7th International Istanbul Smart Grids and Cities Congress and Fair, ICSG 2019 - Proceedings**, p. 31–35, 2019.

CHEN, Qifeng; CHEN, Haoming; CAI, Yanpu; ZHANG, Yanqi; HUANG, Xin. Denial of Service Attack on IoT System. **Proceedings - 9th International Conference on Information Technology in Medicine and Education, ITME 2018**, p. 755–758, 2018.

CISCO. **Cisco Annual Internet Report (2018–2023) White Paper**. 2020. 124 p. Disponível em: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

COSTA, Bruno Sielly J.; BEZERRA, Clauber G.; De Oliveira, Luiz Affonso H.G. A multistage fuzzy controller: Toolbox for industrial applications. **2012 IEEE International Conference on Industrial Technology, ICIT 2012, Proceedings**, IEEE, p. 1142–1147, 2012.

De Donno, Michele; TANGE, Koen; DRAGONI, Nicola. Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog. **IEEE Access**, IEEE, v. 7, p. 150936–150948, 2019. ISSN 21693536.

DEOGIRIKAR, Jyoti; VIDHATE, Amarsinh. Security Attacks in IoT : A Survey. **2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)**, v. 1-, p. 32–37, 2017.

DIRO, Abebe; REDA, Haftu; CHILAMKURTI, Naveen; MAHMOOD, Abdun; ZAMAN, Noor; NAM, Yunyoung. Lightweight Authenticated-Encryption Scheme for Internet of Things Based on Publish-Subscribe Communication. **IEEE Access**, IEEE, v. 8, p. 60539–60551, 2020. ISSN 21693536.

DULIK, Miroslav. Network attack using TCP protocol for performing DoS and DDoS attacks. **2019 Communication and Information Technologies Conference Proceedings, KIT 2019 - 10th International Scientific Conference**, Armed Forces Academy of Gen. M. R. Stefanik, 2019.

EMA, Romana Rahman; ISLAM, Tajul; AHMED, Md Humayan. Suitability of Using Fog Computing Alongside Cloud Computing. **2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019**, IEEE, p. 2019–2022, 2019.

ESPINOSA, Jairo; VANDEWALLE, Joos. Constructing fuzzy models with linguistic integrity from numerical data-AFRELI algorithm. **IEEE Transactions on Fuzzy Systems**, v. 8, n. 5, p. 591–600, 2000. ISSN 10636706.

FIRDOUS, Syed Naeem; BAIG, Zubair; VALLI, Craig; IBRAHIM, Ahmed. Modelling and evaluation of malicious attacks against the IoT MQTT protocol. **Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017**, v. 2018-Janua, p. 748–755, 2018.

G. GEORGE COULOURIS, JEAN DOLLIMORE, TIM KINDBERG, GORDON BLAIR. **Distributed Systems: Concepts and Design**. 5rd edition. ed. *[S.l.]*: Pearson, 2012. ISBN 0-13-214301-1.

HANDA, Arun. Introduction to IMS. **System Engineering For IMS Networks**, p. 1–24, 2018.

HARIPRIYA, A. P.; KULOTHUNGAN, K. Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things. **Eurasip Journal on Wireless Communications and Networking**, EURASIP Journal on Wireless Communications and Networking, v. 2019, n. 1, 2019. ISSN 16871499.

HARSHA, M. S.; BHAVANI, B. M.; KUNDHAVAI, K. R. Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs. **2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018**, IEEE, p. 2244–2250, 2018.

Hernández Ramos, Santiago; VILLALBA, M. Teresa; LACUESTA, Raquel. MQTT Security: A Novel Fuzzing Approach. **Wireless Communications and Mobile Computing**, v. 2018, 2018. ISSN 15308677.

HWANG, Kitae; LEE, Jae Moon; JUNG, In Hwan; LEE, Dong-hee Hee. Modification of Mosquitto Broker for Delivery of Urgent MQTT Message. **2019 IEEE Eurasia Conference on IOT, Communication and Engineering, ECICE 2019**, IEEE, p. 166–167, 2019.

JONES, Joshua; WIMMER, Hayden; HADDAD, Rami J. PPTP VPN: An Analysis of the Effects of a DDoS Attack. **2019 SoutheastCon**, IEEE, p. 1–6, 2020.

JUTADHAMAKORN, Pongnapat; PILLAVAS, Tinnapat; VISOOTTIVISETH, Vasaka; TAKANO, Ryousei; HAGA, Jason; KOBAYASHI, Dylan. A scalable and low-cost MQTT broker clustering system. **Proceeding of 2017 2nd International Conference on Information Technology, INCIT 2017**, v. 2018-Janua, p. 1–5, 2018.

KEPCEOGLU, Bugra; MURZAEVA, Azhar; DEMIRCI, Sercan. Performing energy consuming attacks on IoT devices. **27th Telecommunications Forum, TELFOR 2019**, p. 19–22, 2019.

KEVIN, Asthon. That ' Internet of Things ' Thing. **RFiD Journal**, p. 4986, 2010. ISSN 00280836.

KHANAM, Shapla; AHMEDY, Ismail Bin; Idna Idris, Mohd Yamani; JAWARD, Mohamed Hisham; Bin Md Sabri, Aznul Qalid. A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things. **IEEE Access**, v. 8, p. 219709–219743, 2020. ISSN 21693536.

KHWANRIT, Ruengwit; KITTIPIYAKUL, Somsak; KUDTONAGNGAM, Jasada; FUJITA, Hideaki. Accuracy Comparison of Present Low-cost Current Sensors for Building Energy Monitoring. **2018 International Conference on Embedded Systems and Intelligent Technology and International Conference on Information and Communication Technology for Embedded Systems, ICESIT-ICICTES 2018**, p. 3–8, 2018.

KORONIOTIS, Nickolaos; MOUSTAFA, Nour; SITNIKOVA, Elena. Forensics and Deep Learning Mechanisms for Botnets in Internet of Things: A Survey of Challenges and Solutions. **IEEE Access**, v. 7, p. 61764–61785, 2019. ISSN 21693536.

LEE, Chuen Chien. **Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part I**. 1990. 404–418 p.

LEKIĆ, Milica; GALIĆ, Jovan; MATIĆ, Sonja. An IoT Solution for Secured and Remote Sound Level Monitoring. **2019 18th International Symposium INFOTEH-JAHORINA, INFOTEH 2019 - Proceedings**, v. 1-, n. March, p. 20–22, 2019.

LI, Fangyu; SHI, Yang; SHINDE, Aditya; YE, Jin; SONG, Wenzhan. Enhanced cyber-physical security in internet of things through energy auditing. **IEEE Internet of Things Journal**, IEEE, v. 6, n. 3, p. 5224–5231, 2019. ISSN 23274662.

LIANG, Lulu; ZHENG, Kai; SHENG, Qiankun; HUANG, Xin. A denial of service attack method for an IoT system. **Proceedings - 2016 8th International Conference on Information Technology in Medicine and Education, ITME 2016**, IEEE, p. 360–364, 2017.

LIAO, Zaifei; LU, Xinjie; YANG, Tian; WANG, Hongan. Missing data imputation: A fuzzy k-means clustering algorithm over sliding window. **6th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2009**, IEEE, v. 3, p. 133–137, 2009.

LIN, Jie; YU, Wei; ZHANG, Nan; YANG, Xinyu; ZHANG, Hanlin; ZHAO, Wei. A Survey on Internet of Things : Architecture , Enabling Technologies , Security and Privacy , and Applications. **IEEE Internet of Things Journal (Volume: 4 , Issue: 5 , Oct. 2017)**, v. 4, n. 5, p. 1125–1142, 2017.

MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International Journal of Man-Machine Studies**, v. 7, n. 1, p. 1–13, 1975. ISSN 00207373.

MISHRA, Biswajeeban; KERTESZ, Attila. The Use of MQTT in M2M and IoT Systems: A Survey. **IEEE Access**, v. 8, p. 201071–201086, 2020. ISSN 2169-3536.

MOHAMMED, Yakub; MOHAMMED, Abubakar Saddiq; ABDULKARIM, Hauwa Talatu; DANLADI, Clement; VICTOR, Aduh; EDOKA, Romanus. Development and Implementation of an Internet of Things (IOT) Based Remote Patient Monitoring System. **2019 15th International Conference on Electronics, Computer and Computation (ICECCO)**, IEEE, v. 88, n. Icecco, p. 1–6, 2020.

MURZAEVA, Azhar; KEPCEOGLU, Bugra; DEMIRCI, Sercan. Survey of Network Security Issues and Solutions for the IoT. **3rd International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2019 - Proceedings**, IEEE, v. 1-, n. April 2011, p. 1–6, 2019.

NAIK, Nitin. Fuzzy inference based intrusion detection system: FI-Snort. **Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, 13th IEEE International Conference on Dependable, Autonomic and Se**, IEEE, p. 2062–2067, 2015.

NAZIR, Sajid; KALEEM, Muhammad. Security with MQTT. **2019 International Conference on Information Science and Communication Technology (ICISCT)**, IEEE, p. 1–5, 2019.

OSANAIYE, Opeyemi; CHEN, Shuo; YAN, Zheng; LU, Rongxing; CHOO, Kim Kwang Raymond; DLODLO, Mqhele. From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework. **IEEE Access**, v. 5, p. 8284–8300, 2017. ISSN 21693536.

PALMIERI, Andrea; PREM, Paolo; RANISE, Silvio; MORELLI, Umberto; AHMAD, Tahir. MQTTSA: A Tool for Automatically Assisting the Secure Deployments of MQTT Brokers. **2019 IEEE World Congress on Services (SERVICES)**, IEEE, v. 2642-939X, p. 47–53, 2019.

PAN, Jie; ZHANG, Yiwen; WANG, Qingren; YAN, Dengcheng; ZHANG, Wenming. A Novel Fog Node Aggregation Approach for Users in Fog Computing Environment. **Proceedings - IEEE 18th International Conference on Dependable, Autonomic and Secure Computing, IEEE 18th International Conference on Pervasive Intelligence and Computing, IEEE 6th International Conference on Cloud and Big Data Computing and IEEE 5th Cybe**, p. 160–167, 2020.

PAPPIS, Costas P.; SIETTOS, Constantinos I. Fuzzy reasoning. *In*: **Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques**. *[S.l.]*: Search Methodologies. Springer, Boston, MA, 2005. p. 437–474. ISBN 0387234608.

PAVELIC, Marko; BAJT, Vatroslav; KUSEK, Mario. Energy efficiency of machine-to-machine protocols. **2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings**, Croatian Society MIPRO, p. 361–366, 2018.

PEDRYCZ, Witold; GOMIDE, Fernando A. C. **An Introduction to Fuzzy Sets**: Analysis and design (complex adaptive systems). Massachusetts, USA: MIT Press, 1998. 465 p.

PERALTA, Goiuri; IGLESIAS-URKIA, Markel; BARCELO, Marc; GOMEZ, Raul; MORAN, Adrian; BILBAO, Josu. Fog computing based efficient IoT scheme for the Industry 4.0. **Proceedings of the 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics, ECMSM 2017**, IEEE, p. 1–6, 2017.

PIERLEONI, Paola; CONCETTI, Roberto; BELLI, Alberto; PALMA, Lorenzo. Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison. **IEEE Access**, IEEE, v. 8, p. 5455–5470, 2020. ISSN 21693536.

POTRINO, Giuseppe; De Rango, Floriano; SANTAMARIA, Amilcare Francesco. Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker. **IEEE Wireless Communications and Networking Conference, WCNC**, IEEE, v. 2019-April, p. 1–6, 2019. ISSN 15253511.

ROOHI, Ammara; ADEEL, Muhammad; SHAH, Munam Ali. DDoS in IoT: A roadmap towards security countermeasures. **ICAC 2019 - 2019 25th IEEE International Conference on Automation and Computing**, Chinese Automation and Computing Society in the UK - CACSUK, v. 1, n. 1, p. 1–6, 2019.

SANGODOYIN, Abimbola; MODU, Babagana; AWAN, Irfan; Pagna Disso, Jules. An Approach to Detecting Distributed Denial of Service Attacks in Software Defined Networks. **Proceedings - 2018 IEEE 6th International Conference on Future Internet of Things and Cloud, FiCloud 2018**, IEEE, p. 436–443, 2018.

SCIKIT-FUZZY. **SciKit-Fuzzy**. 2020. Disponível em: https://pythonhosted.org/scikit-fuzzy/overview.html.

SELVACHANDRAN, Ganeshsree; QUEK, Shio Gai; LAN, Luong Thi Hong; SON, Le Hoang; Long Giang, Nguyen; DING, Weiping; ABDEL-BASSET, Mohamed; ALBUQUERQUE, Victor Hugo C. A New Design of Mamdani Complex Fuzzy Inference System for Multi-attribute Decision Making Problems. **IEEE Transactions on Fuzzy Systems**, IEEE, v. 6706, n. c, p. 1–1, 2019. ISSN 1063-6706.

SHAH, Babar. Fuzzy Energy Efficient Routing for Internet of Things (IoT). **International Conference on Ubiquitous and Future Networks, ICUFN**, IEEE, v. 2018-July, p. 320–325, 2018. ISSN 21658536.

SHAPSOUGH, Shams; ALOUL, Fadi; ZUALKERNAN, Imran A. Securing Low-Resource Edge Devices for IoT Systems. **2018 International Symposium in Sensing and Instrumentation in IoT Era, ISSI 2018**, IEEE, 2018.

SHI, Yang; LI, Fangyu; SONG, Wen Zhan; LI, Xiang Yang; YE, Jin. Energy audition based cyber-physical attack detection system in IoT. **ACM International Conference Proceeding Series**, 2019.

SIKORA, Marek; GERLICH, Tomas; MALINA, Lukas. On Detection and Mitigation of Slow Rate Denial of Service Attacks. **International Congress on Ultra Modern Telecommunications and Control Systems and Workshops**, v. 2019-Octob, p. 0–4, 2019. ISSN 2157023X.

SINGH, Kaptan; DEEPAK, Dr. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. **Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018**, v. 4, n. 5, p. 642–646, 2019.

SU, Te Jen; WANG, Shih Ming; CHEN, Yi Feng; LIU, Chao Liang. Attack detection of distributed denial of service based on Splunk. **Proceedings of the IEEE International Conference on Advanced Materials for Science and Engineering: Innovation, Science and Engineering, IEEE-ICAMSE 2016**, IEEE, p. 397–400, 2017.

SUGUNA, M.; RAMALAKSHMI, M. G.; CYNTHIA, J.; PRAKASH, D. A survey on cloud and internet of things based healthcare diagnosis. **2018 4th International Conference on Computing Communication and Automation, ICCCA 2018**, IEEE, v. 1-, p. 1–4, 2018.

SURYA KARTIKA, Luh Gede; RINARTHA, Komang; ATMOJO, Yohanes Priyo; Wayan Sukadana, I. Green Monitoring System for Energy Saving in Accommodation Services. **2019 1st International Conference on Cybernetics and Intelligent System, ICORIS 2019**, IEEE, v. 1, n. August, p. 73–78, 2019.

TIAN, Gui Yun. An Intrusion Detection System Against DDoS Attacks in loT Networks. **2020 10th Annual Computing and Communication Workshop and Conference (CCWC)**, p. 1–6, 2020.

TOLDINAS, Jevgenijus; LOZINSKIS, Borisas; BARANAUSKAS, Edgaras; DOBROVOLSKIS, Algirdas. MQTT Quality of Service versus Energy Consumption. **2019 23rd International Conference Electronics**, IEEE, p. 1–4, 2019.

TRUONG, A. Stanford-Clark; L., H. **MQTT for sensor net- works (MQTTs)**. 219. Disponível em: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.pdf.

VISHWAKARMA, Ruchi; JAIN, Ankit Kumar. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. **Telecommunication Systems**, Springer US, v. 73, n. 1, p. 3–25, 2020. ISSN 15729451. Disponível em: https://doi.org/10.1007/s11235-019-00599-z.

XIAO, Yinhao; JIA, Yizhen; LIU, Chunchi; CHENG, Xiuzhen; YU, Jiguo; LV, Weifeng. Edge Computing Security: State of the Art and Challenges. **Proceedings of the IEEE**, v. 107, n. 8, 2019. ISSN 00189219.

XU, Chuan; XIONG, Zhengying; ZHAO, Guofeng; YU, Shui. An Energy-Efficient Region Source Routing Protocol for Lifetime Maximization in WSN. **IEEE Access**, IEEE, v. 7, p. 135277–135289, 2019. ISSN 21693536.

YOUSAF, Roomana; AHMAD, Rizwan; AHMED, Waqas; HASEEB, Abdul. Fuzzy Power Allocation for Opportunistic Relay in Energy Harvesting Wireless Sensor Networks. **IEEE Access**, v. 5, p. 17165–17176, 2017. ISSN 21693536.

ZADEH. "Fuzzy Sets", Information and control. v. 8, p. 338–353, 1965.

ZADEH, L. A. The concept of a linguistic variable and its application to approximate reasoning-I. **Information Sciences**, v. 8, n. 3, p. 199–249, 1975. ISSN 00200255.

ZHAO, Shancang Li & Li Da Xu & Shanshan. The internet of things: a survey. **Springer Science+Business Media New Yor**, 2014.

ZIMMERMANN, H. J. **Fuzzy Set Theory, and its applications**. 4 edition. ed. Dordrecht: Springer Netherlands, 2001. 526 p. ISBN 9789401038706.