



A Fuzzy Inference System for DDoS Identification in Fog Computing based on Energy Consumption

Diogo Vinicius Martins da Cruz  [Universidade Tecnológica Federal do Paraná |


diogocruz@alunos.utfpr.edu.br]

Ana Cristina Barreiras Kochem Vendramin  [Universidade Tecnológica Federal do Paraná |

criskochem@utfpr.edu.br]

Daniel Fernando Pigatto   [Universidade Tecnológica Federal do Paraná | pigatto@utfpr.edu.br]

Juliana de Santi  [Universidade Tecnológica Federal do Paraná | jsanti@utfpr.edu.br]

 Programa de Pós-Graduação em Computação Aplicada (PPGCA), Universidade Tecnológica Federal do Paraná, Av. Sete de Setembro, 3165, Rebouças, Curitiba, PR, 80230-901, Brazil.

Received: 05 November 2024 • Accepted: 01 May 2025 • Published: 21 May 2025

Abstract Internet of Things (IoT) networks, characterized by their heterogeneous devices, standards, and features, along with limited energy resources, are particularly vulnerable to security threats. Fog computing, which processes data closer to the network edge (i.e., IoT devices), has emerged as a key paradigm for addressing these issues. The Message Queuing Telemetry Transport (MQTT) protocol is commonly used for communication between IoT and fog devices due to its simplicity and ease of implementation. However, MQTT does not include built-in security measures, making it susceptible to Distributed Denial of Service (DDoS) attacks. This paper identifies the main DDoS threats in the context of the MQTT protocol and proposes a Fuzzy Inference System (FIS) designed to detect and classify specific DDoS attack types. By analyzing energy consumption patterns in fog nodes, fuzzy logic infers the degree of membership of a DDoS attack in a fog node, providing a robust method for threat detection in IoT environments.

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

1 Introduction

The Internet of Things (IoT) revolutionizes various sectors by connecting devices and systems, enabling real-time data exchange and automation. Innovations in IoT are driving advancements in sectors like smart cities, precision agriculture, healthcare, industrial automation, and consumer electronics, enhancing efficiency and convenience. Consequently, equipment with functional intelligence has increasingly filled important roles in daily life, influencing behavior and enhancing quality of life while saving time. Furthermore, projections suggest that the number of IoT devices will surpass 32.1 billion by 2030, nearly doubling from 15.9 billion in 2023 [Vailshery, 2024].

The demands of this pervasive computing environment include real-time data processing and mobility, making the Fog Computing (FC) model a suitable solution [Hazra *et al.*, 2023]. Positioned strategically between IoT devices and Cloud Data Centers (DCs) in a layered hierarchical structure, FC provides cloud-like services closer to the network edge, which ensures low latency, faster response times, and decreased bandwidth consumption [Mathur *et al.*, 2023]. However, integrating FC with IoT systems introduces new security challenges due to the increased interactions and heterogeneity of the system, demanding measures aligned with the CIA (Confidentiality, Integrity, and Availability) triad [Damerdjı *et al.*, 2024].

In recent years, there has been a significant increase in research focused on the identification, prevention, and mitigation

of threats to IoT systems [Al-Fayoumi and Abu Al-Haija, 2023; Damerdjı *et al.*, 2024]. Although IoT architecture is structured into multiple layers, many attack mitigation solutions focus on the higher levels, often neglecting the lower layers [Butun *et al.*, 2020]. Given the physical resource constraints of IoT devices, managing energy consumption is a critical aspect of incorporating security measures into this ecosystem [Firdous *et al.*, 2017].

MQTT (Message Queuing Telemetry Transport) is one of the primary communication protocols used in fog computing [Andy *et al.*, 2017], especially suitable for resource-limited devices. However, its simplicity, along with poor implementation practices, exposes IoT ecosystems to significant security risks. In [Lakshminarayana *et al.*, 2024], a comprehensive review of attacks and countermeasures for securing MQTT-based IoT networks is provided. Among these, Distributed Denial of Service (DDoS) attacks, particularly volumetric or flooding attacks, are the most prevalent in MQTT networks [Vishwakarma and Jain, 2020; Lakshminarayana *et al.*, 2024].

DDoS-induced packet flooding forces the IoT/Fog target to process an excessive number of packets, thus draining its already constrained energy resources [Roohi *et al.*, 2019; Kepceoglu *et al.*, 2019]. Volumetric attacks in MQTT networks primarily target the central devices within this protocol's architecture, specifically the MQTT brokers. These brokers operate within the Fog computing layer and are classified as Fog Nodes [Haripriya and Kulothungan, 2019].

Most MQTT brokers support encryption and access con-

trol mechanisms, such as authentication and Access Control Lists (ACLs), which are essential for preventing unauthorized clients from publishing or subscribing to specific topics. These mechanisms ensure that only authenticated users can access sensitive data and participate in communication. Although brokers are generally more robust than edge devices (i.e., IoT devices), their essential functions mean that their temporary disabling or interruption can significantly impact connected devices [Vishwakarma and Jain, 2020; Lakshminarayana *et al.*, 2024].

The detection of an attack based on the network traffic analysis is something very common in the literature. However, this approach is not always feasible, as distinguishing between normal traffic and attack traffic can sometimes be challenging [Shukla *et al.*, 2024]. Sophisticated attacks often exhibit subtle patterns or mimic legitimate traffic, increasing the risk of false negatives [Tian, 2020; Shukla *et al.*, 2024].

Effectively analyzing network traffic becomes difficult when facing zero-day or unknown attacks, as their unpredictable and novel behavior makes accurate evaluation and detection problematic [Shukla *et al.*, 2024].

In some cases, IoT nodes may generate unknown traffic patterns. Relying solely on this type of data can lead to an increase in false positives during flooding attack detection [Roohi *et al.*, 2019]. Additionally, while some detection solutions rely on system logs, these logs can be easily forged or their analysis may be overlooked [Li *et al.*, 2019]. On the other hand, real-time analysis of massive network traffic can overwhelm detection systems [Shukla *et al.*, 2024].

Our proposed method for identifying DDoS attacks addresses several gaps present in existing models by analyzing the energy consumption of an MQTT broker. This approach offers a viable solution for detecting volumetric attacks in Fog computing environments and provides insights into how each type of attack impacts its target through the use of fuzzy logic. Unlike conventional binary logic, fuzzy logic allows for a more nuanced representation of each DDoS attack and its degree of membership. It effectively models non-linear and complex systems, akin to human reasoning, and reduces reliance on complex mathematical models [Shah *et al.*, 2015; Berouine *et al.*, 2019; Javaheri *et al.*, 2023]. The membership values detected by a Fuzzy Inference System (FIS) and the identification of the attack modality facilitate rapid decision-making, even in the case of less severe attacks.

The remainder of this paper is structured as follows. Section 2 reviews the state of the art that motivated this study. Section 3 discusses the characteristics and limitations of IoT environments operating with Fog Computing, focusing on relevant protocols and security issues, including DDoS attacks. Section 4 presents the concepts of fuzzy logic. Section 5 details the proposed architecture and FIS designed for measuring energy consumption and identifying DDoS attacks in a fog computing environment. Section 6 presents the experimental results. Section 7 summarizes the conclusions and outlines future directions.

2 Related Work

This section presents some strategies that closely align with the objectives of our study, including comparing different DDoS attack techniques to evaluate their impact on target systems; analyzing energy consumption in Fog hardware with resource limitations; detecting and mitigating DDoS attacks on MQTT brokers; applying fuzzy logic in Fog and IoT contexts, particularly for efficient energy consumption; and utilizing fuzzy logic in cyber attack scenarios.

Many types of threats in the IoT environment are grouped and quantified in [Deogirikar and Vidhate, 2017], where the most harmful threat is identified through some classification parameters. The paper presents and summarizes a range of IoT attacks, which facilitates new research opportunities and the development of effective countermeasures.

The comparison of denial of service attacks is approached in [Sangodoyin *et al.*, 2018]. In this work, an Intrusion Detection System (IDS) is developed for a Software Defined Networking (SDN) environment, where the devices are subjected to three different types of Denial of Service (DoS) attacks: SYN flooding, ACK flooding, and HTTP flooding. An attack is identified through a significant change in network latency, observed in a comparison between an attacked and a non attacked environment. An important point highlighted by the authors is that the detection of volumetric attacks carried out over long periods may be inaccurate if based solely on traffic analysis.

[Shukla *et al.*, 2024] provides a review of DDoS attacks in IoT-environment, with a focus on classification by perception, network, and application layers. The paper analyzes the existing Machine Learning (ML) and Deep Learning (DL)-based detection approaches for large-scale IoT traffic-based DDoS attacks. It also emphasizes the drawbacks of existing DDoS datasets, such as their outdated nature, unbalanced distribution between malicious and benign traffic, and limited representation of real-world network environments.

In [Damerdjji *et al.*, 2024], the security challenges of using Fog Computing architecture in IoT are analyzed, with particular emphasis on authentication, confidentiality, and data integrity. The paper highlights common attacks against these security measures in a Fog-IoT environment and reviews various proposed security mechanisms.

In [Kepceoglu *et al.*, 2019], the energy consumption of IoT devices during DDoS attacks is analyzed, providing data on the resource consumption of hardware at the attack targets. In a Local Area Network with four connected devices, two types of DDoS attacks, SYN Flooding and ICMP Flooding, are executed. Considering changes in the energy consumption of a Raspberry Pi, a method for detecting various modalities of physical-damage cyberattacks is proposed in [Shi *et al.*, 2019]. The data is processed in two phases. The first phase detects anomalies in the consumption pattern, while the second phase identifies the type of attack based on the level of energy consumed. The study emulates two of the most common attacks: denial of service and overheating, along with others such as viruses, intrusions, Trojan horses, and blackouts. The time required to detect the attack is 180 seconds.

The solution proposed by [Li *et al.*, 2019] addresses the de-

tection of physical and cyber attacks through energy auditing. The solution utilizes multiple Raspberry Pi units, which are subjected to both physical and cyber attacks. Physical attacks are conducted through external overheating, while DDoS attacks represent the cyber component. The system first identifies the energy consumption patterns and then alerts users to any detected changes. The input metrics include processor, network interface, and hard drive consumption. The results are categorized into three types: Physical Attack, Cyber Attack, and Non-Attack.

The analysis of MQTT protocol threats is discussed in [Firdous *et al.*, 2017], which highlights DDoS attacks as the most significant threat. To illustrate the importance of the broker, its resilience under two types of DDoS attacks (SYN flooding and large packet flooding) is evaluated in [Firdous *et al.*, 2017]. The study examines changes in memory and CPU usage associated with each attack, using a virtual machine running the *hping3* tool on the Kali Linux operating system.

A taxonomy for understanding MQTT vulnerabilities, classifying the attacks that exploit these weaknesses, and outlining the defense mechanisms designed to mitigate them is provided in [Lakshminarayana *et al.*, 2024]. The authors also highlights the lack of realistic datasets for training intrusion detection models, and emphasizes the need for lightweight solutions adapted to the resource constraints of most IoT devices.

A lightweight strategy for detecting Low-Rate DDoS (LR-DDoS) attacks against the MQTT protocol in a Software-Defined IoT (SD-IoT) context, employing machine learning models with a focus on a minimal feature set, is presented in [Al-Fayoumi and Abu Al-Haija, 2023]. To identify the two critical features necessary to achieve high prediction accuracy and low computational overhead, Principal Component Analysis (PCA) was used. Supervised learning techniques were applied to analyze real-time traffic in an SD-IoT dataset (LRDDoS-MQTT-2022).

DDoS attacks and anomaly detection methods, with a specific focus on fuzzy logic-based techniques, are reviewed in [Javaheri *et al.*, 2023]. The paper also introduces a hierarchy of DDoS attacks based on their internal mechanisms and protocols, and a taxonomy of the fuzzy-based anomaly detection approaches, including fuzzy inference system-based schemes. The authors highlight the necessity for DDoS detection systems with low runtime complexity, adaptable to system changes, and able to detect anomalies even in encrypted traffic.

Focusing on Industrial Internet of Things (IIoT) systems, a three-stage risk analysis model based on fuzzy logic to evaluate the level of security risks in these environments is presented in [Kerimkhulle *et al.*, 2023]. Each stage employs a dedicated fuzzy inference system. In the first stage, the probability of threat occurrence is determined based on three risk criteria: asset significance, existing controls in the system, and historical incidents. The second stage focuses on assessing potential damage, with financial and reputational impacts as the key criteria. Finally, the overall security risk is then computed from the outputs of the first two stages.

Our paper identifies three modalities of volumetric denial of service attacks that target resource exhaustion in an MQTT broker. The proposed method monitors the victim's energy

consumption to detect and identify cyber attacks based on changes in consumption patterns. Fuzzy logic is used to assess the severity of each threat by evaluating its degree of membership. This study aims to enhance decision-making and facilitate the rapid implementation of countermeasures for each type of attack.

Table 1 summarizes the above-mentioned works, showing how our proposed strategy aligns with or diverges from existing approaches.

3 IoT and Fog Computing

The expression Internet of Things (IoT) was created by Kevin Ashton in the end of the 1990s and can be defined as an environment of connected intelligent objects which interact in the real world [Potrino *et al.*, 2019]. IoT explores the common characteristics of these objects, turning them into intelligent things [Yassein *et al.*, 2018].

The intense interaction of things and people provided by IoT influences people's behavior and life quality. However, the interaction between smart equipment may happen with little or none interaction of people, which is known as Machine-to-Machine (M2M) communication [Deogirikar and Vidhate, 2017; Firdous *et al.*, 2017].

The IoT adopts a layered functional division similar to the OSI (Open Systems Interconnection) model and Fog Computing, utilizing a hierarchical organization of tasks. Each layer in this hierarchy has its own vulnerabilities, which can be exploited by cybercriminals [Li *et al.*, 2019; Vishwakarma and Jain, 2020; Shukla *et al.*, 2024].

IoT architecture is commonly presented in three layers: Perception Layer, Network Layer, and Application Layer [Xu *et al.*, 2019]. The perception layer (also known as the Sensor Layer) is composed of physical components of IoT, such as sensors and RFID (Radio Frequency Identification) actuator devices, and is responsible for data capture through radio frequencies [Xu *et al.*, 2019; Aceto *et al.*, 2019]. The Network Layer is responsible for intercommunication and routing of data collected by sensors and is composed by technologies such as Wi-Fi, Zigbee, and 4G. The Application Layer, composed by different protocols (e.g. MQTT), is where the intelligence and architecture business rules are developed, making data that comes from the sensors useful to the end user [Deogirikar and Vidhate, 2017; Lin *et al.*, 2017].

One of the main attributes of IoT devices is its resource limitation. Networks composed by these equipment normally contain sensors and actuators, which are not very structurally robust [Naik Nitin, 2017; Diro *et al.*, 2020]. This characteristic presents advantages such as energy saving and low cost, but it is also largely exploited by attackers [Firdous *et al.*, 2017; Lakshminarayana *et al.*, 2024].

IoT will be boosted by the complete implementation of IPv6 protocol (Internet Protocol version 6), due to the distribution of billions of unique addresses, allowing an exclusive identification for each one of the many types of IoT devices [Deogirikar and Vidhate, 2017]. The constant growth [Vailshery, 2024] and the heterogeneous set of hardware and software solutions that compose the IoT environment favor the

Table 1. Summary of Related Work.

Reference	Focus Area	Approach	Key Findings
[Deogirikar and Vidhate, 2017]	IoT threat classification	Classification of IoT threats and assessment of their impact using specific parameters	Summarizes IoT attacks, supporting new research opportunities and development of countermeasures.
[Sangodoyin et al., 2018]	Denial of Service (DoS) in SDN	Intrusion Detection System (IDS) for SDN to detect SYN, ACK, and HTTP flooding attacks based on network latency	Identifies limitations in detecting prolonged volumetric attacks based solely on traffic analysis.
[Firdous et al., 2017]	MQTT protocol vulnerabilities	Analysis of MQTT broker resilience under SYN and large packet flooding attacks	Assesses memory and CPU usage impacts under DDoS conditions in MQTT.
[Kepceoglu et al., 2019]	Energy consumption during DDoS	Analysis of energy consumption of IoT devices during SYN and ICMP flooding attacks	Provides data on resource usage under DDoS attacks in IoT environments.
[Shi et al., 2019]	Physical damage cyberattack detection	Detection method based on energy consumption anomalies in Raspberry Pi devices	Differentiates attack types (e.g., DoS, overheating) with a detection time of 180 seconds.
[Li et al., 2019]	Detection of physical and cyberattacks	Energy auditing on multiple Raspberry Pi units to detect attack types	Categorizes detected incidents into Physical Attack, Cyber Attack, or Non-Attack based on energy patterns.
[Al-Fayoumi and Abu Al-Haija, 2023]	Low-Rate DDoS detection in MQTT	Uses PCA and supervised learning to detect LR-DDoS attacks in SD-IoT environments	Achieves high prediction accuracy and low computational overhead through minimal feature set.
[Javaheri et al., 2023]	DDoS detection with fuzzy logic	Reviews fuzzy logic-based DDoS detection methods and attack taxonomy	Highlights demand for low-complexity detection adaptable to encrypted traffic and system changes.
[Kerimkhulle et al., 2023]	Risk analysis in IIoT	Three-stage fuzzy logic-based model for security risk evaluation	Calculates overall security risk based on threat probability and potential impact.
[Shukla et al., 2024]	DDoS attacks in IoT	Review of DDoS attacks classified by perception, network, and application layers; analysis of ML/DL-based detection approaches	Highlights the need for updated and balanced DDoS datasets to improve detection in real-world scenarios.
[Damerdji et al., 2024]	Security challenges in Fog Computing	Focuses the discussion on confidentiality, integrity, and availability (CIA) for securing data in Fog computing	Highlights the common attacks against CIA measures in a Fog-IoT environment, along with existing countermeasures.
[Lakshminarayana et al., 2024]	MQTT security taxonomy	Classification of MQTT vulnerabilities and defense mechanisms	Emphasizes need for realistic datasets and lightweight solutions for IoT devices.
This work	DDoS detection and identification in MQTT (IoT-Fog)	A Fuzzy Inference System (FIS) analyzes energy consumption of MQTT broker to detect/identify SYN, ICMP, and Large Packet flood attacks	Operating independently of network traffic analysis, it effectively models complex systems while reducing the dependence on complex mathematical models. This approach provides a detailed representation of each DDoS attack and its membership level, supporting rapid decision-making, even for less severe attacks.

emergence of new vulnerabilities, making it susceptible to attacks [Kepceoglu *et al.*, 2019; Damerdjil *et al.*, 2024; Shukla *et al.*, 2024; Lakshminarayana *et al.*, 2024]. Due to resource constraints, limitations of edge sensors, and the large volume of data to be analyzed, there is a need for an infrastructure that can process data and manage sensitive information in a scalable and timely manner. In this context, Fog Computing has emerged as a solution [Mathur *et al.*, 2023].

Fog Computing integrates the network edge with IoT and cloud computing, acting as an intermediary layer that extends the capabilities of the cloud [Peralta *et al.*, 2017; Osanaiye *et al.*, 2017; Xu *et al.*, 2019; Hazra *et al.*, 2023]. In this architecture, the end IoT devices, positioned at the lowest and most numerous level, communicate with IoT gateways, which act as intermediaries to the fog layer. The fog layer enhances network flexibility and scalability, facilitating communication with the highest level where the cloud servers reside [Peralta *et al.*, 2017]. Figure 1 presents the Fog Computing architecture, which also reflects the proportional distribution of devices across each layer.

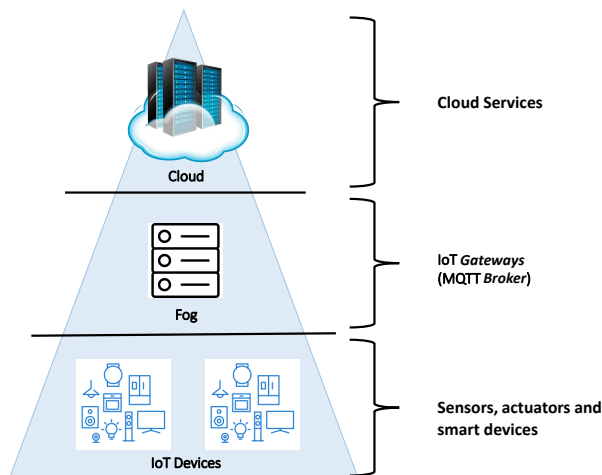


Figure 1. Fog Computing architecture.

The emergence of Fog Computing is partly due to the lack of quality services at the network edge to handle the large volume of data from geographically distributed sources [Osanaiye *et al.*, 2017]. Therefore, Fog Computing brings cloud computing closer to the end IoT devices (edge devices), performing data processing and analysis closer to the source, which enables faster execution of any derived actions [Xu *et al.*, 2019].

Fog Computing architecture is composed by three layers. The outer layer, Internet of Things, is composed by smart “things”, such as sensors and embedded systems [Osanaiye *et al.*, 2017]. The middle layer consists on Fog Nodes, which are devices destined to the analysis and processing of data. The inner layer is the Cloud, with more robust data centers [Xu *et al.*, 2019; Diro *et al.*, 2020].

There are several application layer protocols used in Fog and IoT interactions. The most common protocols are: Message Queue Telemetry Transport (MQTT), Hypertext Transfer Protocol (HTTP), Constrained Application Protocol (CoAP), Extensible Messaging and Presence Protocol (XMPP), and the Advanced Message Queuing Protocol (AMQP) [Nazir and Kaleem, 2019; Potrino *et al.*, 2019;

Kepceoglu *et al.*, 2019; Diro *et al.*, 2020].

In this paper, MQTT is chosen as the protocol due to its widespread use in communication between IoT devices and Fog Computing, especially for devices with limited computing resources, such as actuators and sensors. In the MQTT paradigm, the central component is called a broker. This broker operates within the Fog Layer and is classified as a Fog Node. Fog Nodes are hardware infrastructures that provide resources and services to the edge of the network [Xu *et al.*, 2019; Diro *et al.*, 2020].

MQTT is a lightweight messaging protocol that operates at the application layer and is widely used in Fog Computing [Andy *et al.*, 2017; Firdous *et al.*, 2017; Al-Fayoumi and Abu Al-Haija, 2023; Lakshminarayana *et al.*, 2024]. The MQTT protocol was designed for networks with low bandwidth, ensuring reliable communication while operating over a TCP/IP (Transmission Control Protocol/Internet Protocol) stack. The standard MQTT port number is 1883/TCP for regular messages or 8883/TCP using SSL/TLS cryptography (Secure Sockets Layer/Transport Layer Security) [Naik Nitin, 2017; Firdous *et al.*, 2017; Lekić *et al.*, 2019].

In this protocol, clients cannot forward messages directly to one another; instead, all messages must pass through the broker, due to the adoption of the Publish and Subscribe paradigm [Firdous *et al.*, 2017]. The Publish and Subscribe Paradigm is an alternative to the traditional client/server model, which is seen in HTTP. HTTP is an application layer protocol that operates in 80/TCP port, consistently used in web applications, with the downside of overcharge when used in limited devices such as IoT devices [Naik Nitin, 2017; Diro *et al.*, 2020]. In the client/server model, a device can send messages directly to the end client. In the publish/subscribe paradigm, regardless of the protocol used, the publisher (the device sending the message) and the subscriber (the device receiving the message) are completely decoupled. This decoupling means that the sender does not need to know the receiver’s details (such as IP address, port, or location), and they do not need to be active simultaneously to exchange messages [Diro *et al.*, 2020]. Instead, an MQTT broker is responsible for managing communication and ensuring synchronization between these two independent and unknown endpoints.

The broker is the central component in MQTT communication. It is responsible for receiving all messages, filtering them, and determining which devices should receive each message. Additionally, it provides a basic security layer for message exchange through user authentication. In the Publish/Subscribe paradigm, message exchange between processes is not possible without an intermediary broker [Andy *et al.*, 2017; Naik Nitin, 2017; Firdous *et al.*, 2017; Lekić *et al.*, 2019].

Communication in the Publish/Subscribe paradigm occurs through message topics. Topics are hierarchical text strings, separated by “/”, used to organize the Publish and Subscribe processes managed by the broker [Nazir and Kaleem, 2019; Toldinas *et al.*, 2019]. Publishers create topics when publishing messages, and subscribers specify these topics when subscribing. Only devices subscribed to a particular topic can communicate with the broker regarding that topic [Naik Nitin, 2017; Firdous *et al.*, 2017; Toldinas *et al.*, 2019; Lekić

et al., 2019]. A client device initiates a connection with the broker either by subscribing to a topic or by publishing messages. In summary, the publish/subscribe paradigm operates as follows:

1. The publisher publishes messages in a topic on the MQTT broker;
2. The subscriber connects to the broker subscribing to a message topic;
3. The MQTT broker forwards the message to all topic subscribers.

MQTT is designed to use minimal bandwidth and operate over unreliable networks while requiring minimal implementation effort from developers. The message format consists of a fixed-size header (2 bytes) and an optional header that includes payload and UTF-8 encoded data [Palmieri *et al.*, 2019]. MQTT supports three Quality of Service (QoS) levels—0, 1, and 2—that define the reliability of message delivery [Potrino *et al.*, 2019]. Both the subscriber and publisher can specify a QoS level, and if they differ, the broker defaults to the lower QoS level. The QoS levels and their characteristics are [Toldinas *et al.*, 2019]:

- **At most once:** known as “fire and forget”. This is the default MQTT QoS and is represented by the number “0”. It works with the minimum effort approach which does not ensure message delivery, does not generate feedback signal and does not store the message for a possible resend;
- **At least once:** the QoS level 1 ensures that a message is delivered at least once to the recipient. The sender stores the message until it receives a PUBACK packet from the recipient, which ensures the message reception. In this QoS it is possible that a message is sent or delivered many times, in case of resending happening before confirmation;
- **Exactly once:** the QoS level 2 ensures that a message is delivered exactly once, ensuring that duplicity of message reception by the recipient (subscribe) does not occur. In order to ensure the reliability of this QoS it is necessary to send two pairs of reception confirmation, known as *four-part handshake* which happens in both ways for all sent messages. This duality in confirmation between sender and receiver ensures the precision of message reception only once, while reception duplicity does not occur (as seen in QoS 1). While the reception of a message is not confirmed by the receiver it is not discarded in the sender.

Higher QoS levels increase energy consumption, potentially doubling it when using QoS 2 compared to QoS 0 (the extremes) [Toldinas *et al.*, 2019]. This can be a significant issue, given that IoT devices are typically constrained by low energy consumption, limited memory, and minimal processing capacity. Additionally, while the MQTT protocol supports some security features, these can add considerable overhead to communication [Firdous *et al.*, 2017].

Denial of Service (DoS) attacks, in all their variations, are the most common threats exploiting vulnerabilities in the MQTT protocol [Firdous *et al.*, 2017; Brun *et al.*, 2019; Tian,

2020]. There are two types of Denial of Service attacks when classified from the source of attacks. Malicious attacks that come from a single source are named Denial of Service (DoS) attacks [Haripriya and Kulothungan, 2019]. On the other hand, attacks that originate from multiple sources are known as Distributed Denial of Service (DDoS) attacks [Kepceoglu *et al.*, 2019]. The attacks are also classified according to the impact in resource, bandwidth, network infrastructure, and others, demonstrating a great variation of methods and scope of acting [Brun *et al.*, 2019; Roohi *et al.*, 2019].

Usually, DoS/DDoS attacks aim at disabling the network resources, promoting delay or interruption of a service to valid users or consuming hardware resources from a target [Sikora *et al.*, 2019; Shi *et al.*, 2019]. In Fog Computing environments using the MQTT protocol, a volumetric DDoS attack can severely disrupt data traffic between processes due to the critical role of the broker. Such attacks are designed to consume the physical resources of this central device [Vishwakarma and Jain, 2020]. Among the various volumetric DDoS attack types, the following are particularly relevant to MQTT environments:

- **SYN Packet Flood Attack:** A TCP SYN Flood attack exploits vulnerabilities in the TCP protocol’s three-way handshake. When a host receives a connection request via a SYN packet, it responds with a SYN-ACK. The attacker takes advantage of this process by sending numerous SYN requests to the victim but never responding with the ACK to the SYN-ACK messages. This flood of requests overwhelms the target by exceeding its capacity to handle incoming connections, filling its TCP connection table and thereby hindering or preventing new legitimate connections [Firdous *et al.*, 2017; Brun *et al.*, 2019; Nazir and Kaleem, 2019; Dulik, 2019];
- **Large Packet Flooding:** since MQTT allows a payload of up to 256 MB, malicious users may send numerous messages that exceed this limit in order to consume the broker’s resources [Nazir and Kaleem, 2019]. In this attack, the victim is flooded by fragmented packets. This technique can be further exacerbated by using high QoS levels for messaging, which can cause even more damage to the broker [Potrino *et al.*, 2019];
- **ICMP Packet Flood Attack:** ICMP is a protocol typically used to generate error messages about the inaccessibility of a destination. In an ICMP Flood attack, this behavior is exploited by overwhelming the target with a large volume of ICMP messages [Kepceoglu *et al.*, 2019]. The attacker sends numerous echo request packets without waiting for the corresponding echo reply from the target. The flood occurs due to the large volume of packets sent in a short period of time [Vishwakarma and Jain, 2020].

4 Fuzzy Logic

The Fuzzy Set Theory is intended to model uncertainties and its functions are applied in the most different areas [Zadeh, 1975]. In Fuzzy, both the subjectivity of a given data and the experience of professionals (specialists) are considered.

It operates with non-linear mapping and handles imprecise and conflicting values, resembling human logic. Fuzzy logic is not binary like the conventional logic. It allows for an infinite range of intermediate values between two endpoints [Pedrycz and Gomide, 1998; Shah *et al.*, 2015].

A Fuzzy Inference System (FIS) is comprised of four units (see Figure 2): Fuzzification, Inference (Reasoning), Knowledge, and Defuzzification [Zadeh, 1975; Jang *et al.*, 1997; Pedrycz and Gomide, 1998; Bougdira *et al.*, 2019; Zimmermann, 2001].

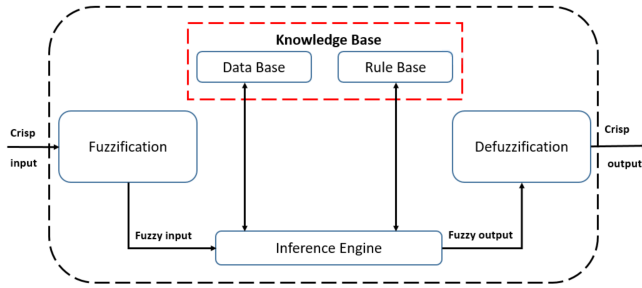


Figure 2. Fuzzy Inference System.

4.1 Fuzzification Unit

The inputs of a FIS are precise values, also called crisp values. In the fuzzification unit the precise input values are converted into fuzzy sets, according to their membership functions [Zadeh, 1975; Shah *et al.*, 2015; Yousaf *et al.*, 2017]. The most common membership functions are trapezoidal, Gaussian, and triangular [Liao *et al.*, 2009].

The input linguistic variables are defined in the fuzzification unit. The linguistic variables represent concepts (e.g. “temperature”, “weight”, etc.), accept linguistic terms (e.g. “slightly”, “a lot”, “little”, etc.) and logical connectors (e.g. “AND”, “OR” and “NO”) [Bougdira *et al.*, 2019; Costa *et al.*, 2012; Mamdani, 1974].

Linguistic variables are structures formed by the Universe of Discourse. The universe of discourse is defined by the maximum and minimum limits of values a variable can take and is composed of linguistic terms. A linguistic term is an expression that names the various characteristics a variable can assume [Zadeh, 1975; Pedrycz and Gomide, 1998; Costa *et al.*, 2012; Bougdira *et al.*, 2019].

The membership function of a linguistic variable graphically represents a fuzzy set [Bougdira *et al.*, 2019]. The degree of membership, also referred to as the degree of association, quantifies how strongly an element from the universe of discourse (X) belongs to a fuzzy set (N). It is expressed as a value within the closed interval [0, 1], and is formally defined by the mapping $N: X \rightarrow [0, 1]$. A membership value of 1 indicates full inclusion in the fuzzy set, 0 indicates no membership, and values between 0 and 1 represent partial membership, capturing the inherent uncertainty or imprecision present in real-world data.

At the output of a FIS, represented in a Cartesian plane, the x-axis corresponds to the universe of discourse, while the y-axis represents the degree of membership [Pedrycz and Gomide, 1998; Selvachandran *et al.*, 2019]. These degrees

of membership are typically associated with linguistic categories such as ‘low’, ‘medium’, or ‘high’, enabling the FIS to model complex reasoning in a human-like manner.

Each precise input value must be compared with the linguistic variables defined in the fuzzification unit to determine the degree of membership. This process is known as antecedent matching [Pedrycz and Gomide, 1998; Haripriya and Kulothungan, 2019].

4.2 Knowledge Base

The Knowledge Base consists of two fundamental components: the Rule Base and the Database [Zimmermann, 2001; Pappis and Siettos, 2005; Costa *et al.*, 2012]. The Rule Base comprises the rules defined by experts or derived from empirical analysis. These rules are structured using logical conditionals such as “If,” “If-Then,” “Else,” and “Else-If.” [Zadeh, 1975; Xiaoyan, 1996; Suguna *et al.*, 2018; Selvachandran *et al.*, 2019]. The database contains the numerical definitions required to establish the membership functions used by the fuzzy rule set [Zimmermann, 2001; Bougdira *et al.*, 2019].

4.3 Inference Engine

The Inference Engine generates fuzzy sets for the Defuzzification unit. It processes information from both the Knowledge Base and the fuzzy sets produced by the Fuzzification unit. The input parameters, derived from fuzzification, are known as the aggregation of antecedents [Pedrycz and Gomide, 1998; Zimmermann, 2001; Bougdira *et al.*, 2019]. The result from the aggregation of the antecedent variables can be obtained using either the maximum or minimum function, depending on the operator used: the s-norm (OR) for the maximum function and the t-norm (AND) for the minimum function, as described below [Pedrycz and Gomide, 1998; Espinosa and Vandewalle, 2000; Zimmermann, 2001; Pappis and Siettos, 2005]:

- **s-norm:** $\mu A \text{ OR } \mu B = \max(\mu A, \mu B)$
- **t-norm:** $\mu A \text{ AND } \mu B = \min(\mu A, \mu B)$

Each rule can be activated through an activation rule semantics [Lee, 1990; Zimmermann, 2001; Selvachandran *et al.*, 2019]. The Mamdani [Mamdani and Assilian, 1975] and Takagi-Sugeno [Takagi and Sugeno, 1985] models are the two existing activation semantics. The activation rule semantics inferred by the Mamdani conjunction, given by the operator \min ($\int = \int m = \min$), outputs only the rules with a degree of activation greater than zero [Mamdani and Assilian, 1975; Pappis and Siettos, 2005; Shah *et al.*, 2015; Haripriya and Kulothungan, 2019; Selvachandran *et al.*, 2019]. The final action of the Inference Engine is the aggregation of the activated rules, also known as the aggregation of the consequents [Pedrycz and Gomide, 1998; Selvachandran *et al.*, 2019].

4.4 Defuzzification Unit

The Defuzzification unit produces a single numerical output derived from the fuzzy sets generated by the inference mechanism. Common defuzzification methods include the centroid

method and the mean of the maximums [Pappis and Siettos, 2005; Yousaf *et al.*, 2017; Bougdira *et al.*, 2019; Espinosa and Vandewalle, 2000; Zimmermann, 2001].

5 DDoS Identification in Fog Computing with Fuzzy Decision Making

This Section presents the proposed architecture and how the proposed Fuzzy Inference System works in order to detect and identify DDoS attacks in Fog Computing.

5.1 The Proposed Architecture

The architecture illustrated in Figure 3 is proposed to evaluate the energy consumption of the broker device and verify the effectiveness of the proposed FIS (Section 5.2) in identifying DDoS attacks within a fog computing environment. This architecture consists of the following elements: five clients (one publisher and four subscribers), a broker, a current measurement sensor, a microcontroller, and a Fuzzy Inference System (FIS) designed to detect and identify DDoS attacks in the broker.

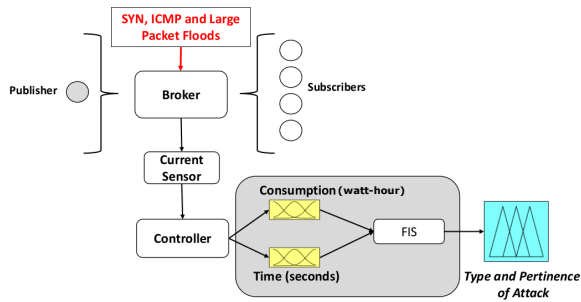


Figure 3. Proposed Architecture.

The three DDoS attack modalities (SYN, ICMP, and Large Packet Floods) were generated separately during the exchange of messages between clients and the broker.

The energy consumed by the broker is read by the sensor before and during each DDoS attack. The collected electric current data (in mAh) is sent to the microcontroller. The function of the microcontroller is to measure the electrical energy consumption, providing both the average energy consumption (in watt-hour (Wh)) and the elapsed time interval (in seconds). These data constitute the inputs for the Knowledge Base of the FIS.

5.2 The Fuzzy Inference System

Both the average energy consumption of the broker and the elapsed time interval serve as input variables of our FIS. The universe of discourse of the variable “consumption” was defined as ranging from 0 to 3 Wh, corresponding to the nominal operating limits of the device used in the broker implementation (Section 6). The variable “time” has an universe of discourse between 0 and 60 seconds, representing a sufficient duration to accurately detect and identify a DDoS attack and its associated degree of membership. The DDoS attack

type represents the FIS output variable, while the degree of membership is a by-product of the FIS output.

In the fuzzification unit, the input (antecedents) linguistic variables are defined as “consumption” and “time”. Each input linguistic variable has five linguistic terms (Consumption: Low Consumption (LC), Medium Consumption (MC), High Medium Consumption (HMC), High Consumption (HC), Very High Consumption (VHC); Time: Very Short Time (VST), Short Time (ST), Medium Time (MT), Long Time (LT), Very Long Time (VLT)). In the construction of the rule base, the Cartesian product of the fuzzy sets for “Consumption” and “Time” (5×5) yields 25 unique combinations, each corresponding to a distinct fuzzy rule. These rules are subsequently stored in the FIS Rule Base, as represented in Table 2.

The output variable output (consequential) consists of four linguistic terms (Attack Identification: No Attack (NA), Large Packet Flood (LP), ICMP Flood (ICMP) e SYN Flood (SYN)). The output of the FIS represents not only the identification of an attack but also its degree of membership.

The fuzzy sets that define the linguistic variables of time and consumption were created by a specialist. These linguistic variables are graphically represented by a triangular membership function. This triangular function, called $trimf(x, y, z)$, requires three input parameters: x is the lower bound, y is the peak (maximum membership), and z is the upper bound. x and z represent the points at which the membership degree is 0, while y is the point where the membership degree is 1. We defined the following triangular membership functions:

Consumption (watt-hour).

- Low Consumption (LC) = [0; 0; 1.9];
- Medium Consumption (MC) = [1.8; 1.9; 2];
- High Medium Consumption (HMC) = [1.9; 2; 2.1];
- High Consumption (HC) = [2; 2.1; 2.2];
- Very High Consumption (VHC) = [2.1; 2.5; 3].

Time (second).

- Very Short Time (VST) = [0; 0; 15];
- Short Time (ST) = [0; 15; 30];
- Medium Time (MT) = [15; 30; 45];
- Long Time (LT) = [30; 45; 60];
- Very Long Time (VLT) = [45; 60; 60].

The output linguistic variable (consequent) represents the behavior of the broker at the end of the data collection period. It is characterized by the attack identification and its degree of membership. The attack identification variable, represented by the horizontal axis (x) of a Cartesian plane, is defined by four linguistic terms and is described by the following triangular membership functions:

Attack Identification

- No Attack (NA) = [0; 0; 30];
- Large Packet Flood (LP) = [25; 40; 55];
- ICMP Flood (ICMP) = [50; 70; 90];
- SYN Flood (SYN) = [85; 99; 99].

The degree of membership, represented by the vertical (y) axis of a Cartesian plane, appears as a by-product of the output and indicates the intensity with which the attack impacts

Table 2. Proposed Rule Base for the Fuzzy Inference System.

Linguistic Term	VST	ST	MT	LT	VLT
LC	NA	NA	NA	NA	NA
MC	NA	NA	NA	NA	NA
HMC	NA	NA	LP	ICMP	SYN
HC	NA	NA	LP	ICMP	SYN
VHC	NA	NA	LP	ICMP	SYN

the broker. Each of the output linguistic terms can assume values between 0 and 1, reflecting the degree of membership.

Membership Degree of Linguistic Terms of Attack

- membership of (No Attack, Large Packet Flood, ICMP Flood or SYN Flood) = [0; 1];

The rule base of FIS is composed of 25 rules, listed from “R1” to “R25”:

- **R1:** if Low Consumption and Very Short Time, then No Attack;
- **R2:** if Low Consumption and Short Time, then No Attack;
- ...
- **R24:** if Very High Consumption and Long Time, then ICMP Flood Attack;
- **R25:** if Very High Consumption and Very Long Time, then SYN Flood Attack.

Table 2 shows the relationship between the FIS antecedents terms. The intersection (line x column) between these terms forms the 25 rules presented in the FIS. The linguistic output terms represent the result of the combination of the antecedents.

The aggregation between the rules is performed using the logical operator “AND” (t-norm), resulting in the minimum value between the antecedents.

The inference of rule activation is performed using the Mamdani semantics, which produces outputs only for rules with a degree of activation greater than zero. In the second stage, aggregation is carried out only with the activated rules using the maximum (norm-s) operator.

The defuzzification method employed in this study is the centroid approach. The centroid method was chosen due to its ability to provide a single numerical output that represents the center of gravity (or mass) of the fuzzy output set distribution [Yousaf *et al.*, 2017]. This characteristic is crucial for producing a clear and interpretable result from the fuzzy inference process.

Additionally, the labels representing the possible conditions of the broker (NA, LP, ICMP, SYN) are positioned on the “x” axis in ascending order of energy consumption. This ordered arrangement enhances the interpretability of the centroid output, allowing it to function effectively as a fuzzy classifier by facilitating a more intuitive and accurate classification of the broker’s condition based on energy consumption.

6 Results

This section presents the analysis of the broker’s energy consumption in both normal and attack scenarios, as well as

the results obtained from the proposed FIS for identifying a DDoS attack and inferring its degree of membership.

The proposed architecture (Fig. 3) is implemented with the following components:

- Five MQTT clients (publisher and subscribers);
- One MQTT broker, represented by the Mosquitto Server running on a Raspberry Pi device. Mosquitto is a widely used MQTT message broker known for being open source, lightweight, and well-suited for IoT devices [Hwang *et al.*, 2019]. The Raspberry Pi 3 Model B was chosen due to its low cost and its popularity as a single-board computer widely used in IoT implementations with MQTT, mainly performing the broker function [Lekić *et al.*, 2019; Nazir and Kaleem, 2019]. The proposed FIS is specifically designed for this model. However, the system can be adapted to work with both lower and higher models of Raspberry Pi;
- One ACS712 current sensor, recognized as the most accurate in its category, is used for precise current measurements [Khwanrit *et al.*, 2018; Surya Kartika *et al.*, 2019];
- One Arduino Uno microcontroller;
- The proposed Fuzzy Inference System (FIS) to detect and identify a DDoS attack in the MQTT broker.

All experiments were conducted using MQTT with QoS level 0, which was intentionally chosen due to its minimal overhead and lower impact on energy consumption. This configuration was crucial for isolating and observing the energy variation primarily caused by the simulated attacks.

6.1 Analysis of the Broker’s Energy Consumption

The energy consumption (in watt-hour, Wh) of the broker device under different types of DDoS attacks exhibits distinct behavior patterns, as illustrated in Figure 4. The energy consumption profile stabilized between 52 and 54 seconds of exposure to the attack, indicating that the system had reached a steady state. Based on this observation, a 60-second interval was selected for analysis, comparing the broker’s energy consumption during normal operation (NA) with its consumption under each of the three DDoS attack types. Therefore, extending the test duration would not yield additional insights into the system’s consumption behavior.

Results were obtained from 10 runs for each condition imposed on the broker (NA, LP, ICMP, SYN). Each point on the graph (with intervals of every 10 seconds) represents the simple arithmetic mean of 10 runs conducted under the same parameters. Based on the standard deviation for 10 runs, we concluded that additional samples would not bring significant variations in the results.

The severity of an attack is directly related to the energy consumed by the broker. Consequently, by analyzing the energy consumption during each attack, the FIS can identify the type of DDoS attack and determine its degree of membership.

Flooding with large packets (Fig.4 - yellow line) was the mildest type of attack, resulting in an 8.02% increase in the broker’s energy consumption after 60 seconds. The broker’s

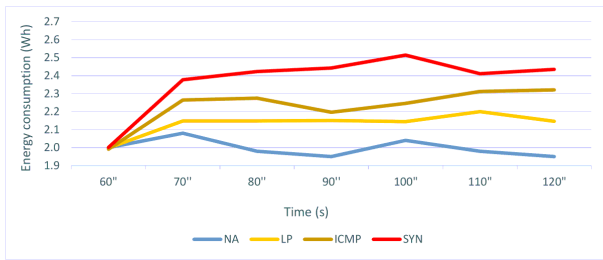


Figure 4. Broker's energy consumption over time

average energy consumption during normal operation (Fig.4 - blue line) was 1.87 watt-hour, rising to 2.02 Wh during the large packet flood attack.

ICMP packet flooding (Fig.4 - olive line) had a moderate impact compared to the other DDoS attack types. After 60 seconds of flooding, this attack increased the broker's energy consumption by 19.3%, reaching an average consumption of 2.13 Wh.

During the SYN packet flood attack (Fig.4 - red line), the energy consumption peaked at 2.52 Wh, with an average consumption of 2.45 Wh. This attack type resulted in the largest difference (31.01%) in energy consumption between the "attack" and "non-attack" scenarios.

6.2 Identification of a DDoS Attack and its Degree of membership

To show that our proposed FIS is able to detect a DDoS attack, identify its type and infer its degree of membership in a MQTT broker, we consider a hypothetical scenario where this broker registers, during 56 seconds, an average energy consumption of 2.1 watt-hour.

Considering these energy consumption and time data, the Algorithm 1 presents the pseudocode of the fuzzyfication unit. Lines 2 and 3 presents the crisp input values: average energy consumption (*consu_input*) and time (*time_input*), respectively. The Universe of Discourse of the consumption (*uni_consu*), time (*uni_time*) and attack (*uni_typeattack*) variables are created in lines 6, 7 and 8, respectively.

The variable *uni_typeattack* = [0; 99] defines the universe of discourse for the output variable in the FIS, representing the classification of the type of attack. Similar to the input variables *uni_consu* (consumption) and *uni_time* (time), it spans a numerical range used for defuzzification and output visualization. The range is divided into 100 discrete points, chosen empirically to provide a high level of granularity, resulting in a smooth and interpretable output surface. This choice offers a balanced trade-off between visual resolution and computational efficiency, avoiding the coarseness of fewer points and the unnecessary complexity of an excessively fine grid.

Algorithm 2 presents the creation of the linguistic variables for consumption (*cunsuLow*, *cunsuMedium*, *cunsuMediumHigh*, *cunsuHigh*, *cunsuVeryHigh*) in lines 2 to 6, time (*veryShortTime*, *shortTime*, *mediumTime*, *longTime* and *veryLongTime*) in lines 9 to 13, and attack (*noAttack*, *floodlp*, *floodicmp* and *floodsyn*) in lines 16 to 19.

Algorithm 3 presents the antecedent matching. The ob-

```

1 #Input Crisp values;
2 consu_input = 2.1;
3 time_input = 56;
4
5 #Universe of Discourse;
6 uni_consu = [0; 3];
7 uni_time = [0; 60];
8 uni_typeattack = [0; 99];

```

Algorithm 1: Pseudocode of the Fuzzyfication unit: input Crisp values and universe of discourse.

```

1 #Linguistic Variables of Consumption;
2 cunsuLow = [0; 0; 1.9];
3 cunsuMedium = [1.8; 1.9; 2];
4 cunsuMediumHigh = [1.9; 2; 2.1];
5 cunsuHigh = [2; 2.1; 2.2];
6 cunsuVeryHigh = [2.1; 2.5; 3];
7
8 #Linguistic Variables of Time;
9 veryShortTime = [0; 0; 15];
10 shortTime = [0; 15; 30];
11 mediumTime = [15; 30; 45];
12 longTime = [30; 45; 60];
13 veryLongTime = [45; 60; 60];
14
15 #Linguistic Variables of Attack;
16 noAttack = [0; 0; 30];
17 floodlp = [25; 40; 55];
18 floodicmp = [50; 70; 90];
19 floodsyn = [85; 99; 99];

```

Algorithm 2: Pseudocode of the Fuzzyfication unit: linguistic variables.

jective is to obtain the degree of combination between the antecedent variables. This is achieved using the *interp_membership(x, y, z)* Python function from the Scikit-fuzzy library. This function infers the membership level of input parameters by evaluating: the fuzzy set that defines the universe of discourse, the fuzzy set that represents the linguistic variables, and the crisp input value. In order to have some degree of matching, the crisp input value must be contained in the related linguistic variable limits.

Algorithm 3 presents only the *matching* operations that results in values higher than zero. The resulting value of line 3 indicates a maximum degree of matching (1,0) between the *cunsuHigh* and the energy consumption input (*consu_input*) antecedents. However, the value of the average energy consumption (2.1 Wh) has a zero degree of matching with the *cunsuLow*, *cunsuMedium*, *cunsuMediumHigh*, and *cunsuVeryHigh* antecedents. This matching process is also done with the time input variable (56 seconds), where only *longTime* (line 8) and *veryLongTime* antecedents (line 11) have a degree of matching higher than zero with this input variable, presenting a resulting value of 0.26 and 0.73, respectively. The other time antecedent variables (*veryShortTime*, *shortTime* and *mediumTime*) result in a zero degree of matching when compared to the time input value.

Algorithm 4 presents the pseudocode of the antecedent ag-

```

1 #Matching of the Antecedent Consumption
  Variables;
2 ...;
3 HC = intermembership(uni_consu; cunsuHigh;
  consu_input);
4 #HC = intermembership([0; 3]; [2; 2.2];
  2.1);
5 ...;
6 ...;
7 #Matching of the Antecedent Time
  Variables;
8 LT = intermembership(uni_time; longTime;
  time_input);
9 #LT = intermembership([0; 60]; [30; 60];
  56);
10 ...;
11 VLT = intermembership(uni_time; veryLongTime;
  time_input);
12 #VLT = intermembership([0, 60], [45,
  60], 56);

```

Algorithm 3: Pseudocode of the Fuzzyfication unit: antecedent matching.

gregation performed by the inference engine of FIS. This aggregation is performed on the resulting values of the antecedent Matching using the “AND” logic operator (t-norm). The result is the minimum value between the aggregated antecedents. The antecedent aggregation yields 25 combinations, which form the FIS rule base. Algorithm 3 shows that only the *cunsuHigh* energy consumption, the *longTime* and *veryLongTime* time antecedents were affected by the precise input values, resulting in a non-zero matching degree (1.0, 0.26 and 0.73, respectively). Lines 1 and 4 of Algorithm 4 shows the results of the only two antecedent aggregations, out of the 25 generated, that yield values different from zero (0.26 and 0.73, respectively).

```

1 LT_HC = min(LT; HC);
2 #LT_HC = min(0.26; 1.0);
3
4 VLT_HC = min(VLT, HC);
5 #VLT_HC = min(0.73; 1.0);

```

Algorithm 4: Pseudocode of the Inference Engine: antecedent aggregation.

After aggregation, the antecedent variables originate the 25 rules of the FIS. These rules are activated using an activation rule semantics. We use the Mamdani model, given by the minimum (t-norm), as the semantic responsible for the activation of rules. Algorithm 5 presents the pseudocode for rule activation according to the Mamdani model.

All 25 rules are subjected to the activation method. Each rule activation receives the antecedent aggregation result and its corresponding attack linguistic terms, as specified by the intersection (line x column) in Table 2. These terms are mapped to values ranging from 0 to 1.

Only rules R19 and R24 are activated as the input values match the adopted parameters for these rules. Line 1 in Algorithm 5 presents Mamdani semantics acting in rule R19

```

1 R19 = min(LT_HC; floodicmp);
2 #R19 = min(0.26, [0; 1]);
3 ...;
4 R24 = min(LT_VHC; floodicmp);
5 #R24 = min(0.73; [0; 1]);
6 ...;
7 #Aggregation of Active Rules by Maximum;
8 activeRules = max(R19; R24);

```

Algorithm 5: Pseudocode of the Inference Engine: Mamdani rule activation.

and obtaining as a result a fuzzy set with values between 0 to 0.26. Line 4 shows the application of Mamdani semantics in rule R24, producing a fuzzy set with values ranging from 0 to 0.73. Line 8 indicates the ending of the inference engine, characterized by the consequent aggregation step. In this step, the activeRules fuzzy set is obtained using the max operator (s-norm), resulting in a value of 0.73. The graphical representation of the rule activation presented in lines 1 and 4 of Algorithm 5 is shown in Figure 5.

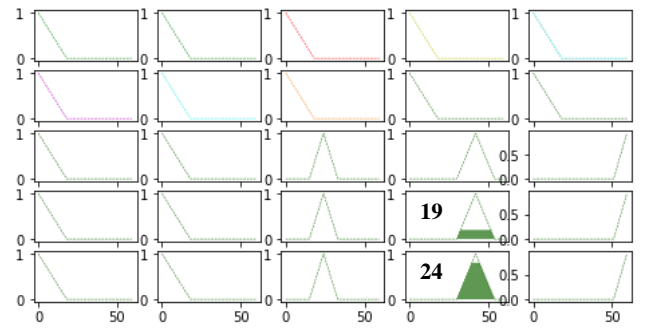


Figure 5. Mamdani activated rules.

The fuzzy set resulting from the aggregation of the active rules (line 8 of Algorithm 5) acts as the input parameter of the defuzzification unit and will determine the membership degree of the attack.

Algorithm 6 shows the pseudocode of the Defuzzification unit. The proposed FIS output, obtained through defuzzification, consists of two precise numeric results. The first result identifies the type of attack, while the second result infers the degree of membership, indicating how significantly the attack affects the broker. The identification of the attack (*id_attack*), given by the *defuzz(x,y,z)* function in *Scikit-fuzzy* Python library, is shown in line 2 of Algorithm 6. This function receives the following input parameters: (i) the universe of discourse of attacks (*uni_typeattack*), representing all types of possible attacks mapped by the FIS; (ii) the resulting set of the active rules aggregation (*activeRules*); and (iii) the defuzzification model used in the study (i.e., the centroid model). The fuzzy set referring to the universe of discourse of attacks (*uni_typeattack*) is an interval of 100 elements [0-99], composed by four sets (noAttack, floodlp, floodicmp, and floodsyn), where each DDoS attack condition is mapped as described in lines 16-19 of Algorithm 2. Therefore, through the Crisp input values (consumption of 2.1 Wh and time of 56 seconds), the function in line 2 of Algorithm 6 obtained the resulting value of 71.14 as an attack identification. This result means an ICMP packet flooding as it is

inserted in the set “ $floodicmp = [50, 70, 90]$ ” (see line 18 of Algorithm 2). Line 6 of Algorithm 6 shows the membership degree ($degree_membership$) of the identified attack. This membership is calculated by the $interp_membership(x,y,z)$ function and receives the following input parameters: (i) the universe of discourse of attacks ($uni_typeattack$); (ii) the resulting set of the active rules aggregation ($activeRules$); and (iii) the previously identified type of attack (id_attack). Considering a broker with an average energy consumption of 2.1 Wh measured for 56 seconds, the proposed FIS indicates an ICMP packet flooding attack ($id_attack = 71.14$) with a membership degree of 0.73 ($degree_membership = 0.73$).

```

1 #Attack Type Identification;
2 id_attack = defuzz(uni_typeattack; activeRules;
  centroid);
3 #id_attack = defuzz([0; 99]; [0; 0.73],
  centroid);
4 ...;
5 #Attack membership;
6 degree_membership = membership(uni_typeattack;
  activeRules; id_attack);
7 #degree_membership = membership([0; 99];
  [0; 0.73]; 71.14);

```

Algorithm 6: Pseudocode of the Defuzzification unit: identification and membership of the attack.

Figure 6 shows the FIS output, with the type of attack displayed on the horizontal axis and the degree of membership on the vertical axis. As observed, the proposed FIS identifies the attack as an ICMP Packet Flooding, with a membership degree of 0.73.

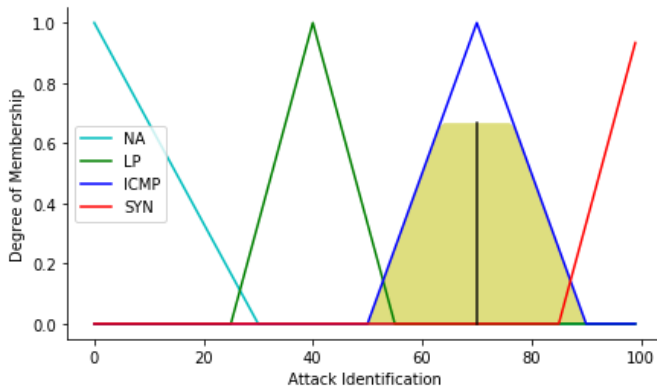


Figure 6. Output of FIS.

7 Conclusion

DDoS attacks are particularly frequent and damaging to devices in Fog Computing environments, especially to broker nodes. Such attacks can lead to brief communication delays, temporary interruptions, or even complete service unavailability. Brokers often run on devices with limited resources, so even a small increase in energy consumption can significantly impact their operation.

This paper proposed a Fuzzy Inference System (FIS) to detect and identify the type of DDoS attack and infer the degree

of membership based on variations in energy consumption patterns. Fuzzy logic is used to handle imprecise and uncertain data, offering more clarity compared to conventional binary logic.

We considered three types of DDoS attacks: SYN packet flooding, large packet flooding, and ICMP packet flooding. These volumetric attacks drain the target’s resources, leading to abnormal energy consumption patterns and degraded performance. In our experiments, each DDoS attack produced unique energy consumption signatures, enabling our FIS to accurately detect and identify the type of DDoS attack affecting the broker. Among the attacks tested, SYN packet flooding generated the highest energy consumption, increasing the broker’s energy use by 31.01% compared to normal operation (i.e., without attacks). This was followed by the ICMP packet flooding attack, which increased the broker’s energy consumption by 19.3%, and the large packet flooding attack, which resulted in an 8.02% increase.

In the proposed scenario, where the broker had an average energy consumption of 2.1 watt-hour over 56 seconds, our FIS successfully identified the type of cyber attack affecting the broker. The output value obtained was 71.14, indicating an ICMP packet flooding attack according to the established fuzzy rules, with a membership degree of 0.73.

7.1 Future Work

In this section, we outline several directions for future work to address current limitations and expand the system’s applicability.

While the system was evaluated in a relatively clean setup with the broker running in isolation, it is important to acknowledge that in real-world scenarios, other services or sensors might be running on the same device, introducing additional variability in energy consumption. To address this, the fuzzy system could be calibrated or adapted for different devices or operational environments. Specifically, the system could be fine-tuned based on device-specific energy profiles to account for the additional load from co-running services, which would help ensure more accurate energy consumption predictions and attack detection. This adaptation would provide greater flexibility, allowing the system to be applied across a wider range of real-world IoT applications, improving its practical applicability.

Another future work will focus on analyzing the energy consumption of fog nodes under various cyberattack scenarios, including cases with unknown and encrypted traffic. The goal is to enhance the fuzzy inference system by incorporating additional features and refining fuzzy rules, ultimately improving the identification, accuracy, and robustness of attack detection and classification.

Although the proposed method was evaluated using real hardware components configured in a physical testbed, where actual energy consumption data were collected under various DDoS attack scenarios, we plan to extend our evaluation to real-world IoT environments to further validate its practical effectiveness.

To evaluate the system’s robustness in more realistic scenarios, it would be important to consider how it behaves under different levels of legitimate traffic, which can vary sig-

nificantly in real-world IoT environments. The fuzzy system could be tested across a range of legitimate traffic volumes, from low to high, to observe how well it distinguishes between legitimate and attack traffic while maintaining accurate attack detection.

Finally, a comparison with simpler strategies, such as static thresholds, shows that while these methods rely on fixed values for attack detection, the fuzzy system provides a more adaptable and dynamic approach. The fuzzy system can adjust to changing traffic patterns and device-specific energy profiles, offering improved accuracy and flexibility in real-world conditions.

Declarations

Acknowledgements

The authors acknowledge the support from the Graduate Program in Applied Computing (PPGCA) at the Federal University of Technology – Parana (UTFPR).

Funding

This research did not receive any funding.

Authors' Contributions

All authors were involved in every stage of the work and contributed equally to its completion.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The source code can be made available by contacting the authors.

References

- Aceto, G., Persico, V., and Pescapé, A. (2019). A survey on Information and Communication Technologies for Industry 4.0: state of the art, taxonomies, perspectives, and challenges. *IEEE Communications Surveys & Tutorials*, 1-(November 2011):1–1. DOI: 10.1109/comst.2019.2938259.
- Al-Fayoumi, M. and Abu Al-Haija, Q. (2023). Capturing low-rate DDoS attack based on MQTT protocol in software Defined-IoT environment. *Array*, 19:100316. DOI: 10.1016/j.array.2023.100316.
- Andy, S., Rahardjo, B., and Hanindhito, B. (2017). Attack scenarios and security analysis of MQTT communication protocol in IoT system. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 4(September):600–604. DOI: 10.11591/eecsi.4.1064.
- Berouine, A., Akssas, E., Naitmalek, Y., Lachhab, F., Bakhouya, M., Ouladsine, R., and Essaaidi, M. (2019). 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*, 1-:1510–1515. DOI: 10.1109/codit.2019.8820356.
- Bougdira, A., Akharraz, I., and Ahaitouf, A. (2019). Fuzzy approach to enhance quality control within intelligent traceability systems. *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS 2019*. DOI: 10.1109/WITS.2019.8723764.
- Brun, O., Yin, Y., Augusto-gonzalez, J., Ramos, M., Brun, O., Yin, Y., Augusto-gonzalez, J., Ramos, M., Gelenbe, E., Attack, I., Brun, O., Yin, Y., Augusto-gonzalez, J., and Ramos, M. (2019). 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*. Available at: <https://laas.hal.science/hal-02062091v1/file/article.pdf>.
- Butun, I., Osterberg, P., and Song, H. (2020). Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. *IEEE Communications Surveys and Tutorials*, 22(1):616–644. DOI: 10.1109/COMST.2019.2953364.
- Costa, B. S. J., Bezerra, C. G., and De Oliveira, L. A. H. (2012). A multistage fuzzy controller: Toolbox for industrial applications. *2012 IEEE International Conference on Industrial Technology, ICIT 2012, Proceedings*, pages 1142–1147. DOI: 10.1109/ICIT.2012.6210094.
- Damerddji, D. O., Lehsaini, M., and Benmahdi, M. B. (2024). A Brief Review on Security in IoT Environments Based on Fog Computing Architecture. In *2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC)*, pages 1–6. DOI: 10.1109/ICEEAC61226.2024.10576289.
- Deogirikar, J. and Vidhate, A. (2017). Security Attacks in IoT : A Survey. *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 1-:32–37. DOI: 10.1109/I-SMAC.2017.8058363.
- Diro, A., Reda, H., Chilamkurti, N., Mahmood, A., Zaman, N., and Nam, Y. (2020). Lightweight Authenticated-Encryption Scheme for Internet of Things Based on Publish-Subscribe Communication. *IEEE Access*, 8:60539–60551. DOI: 10.1109/ACCESS.2020.2983117.
- Dulik, M. (2019). Network attack using TCP protocol for performing DoS and DDoS attacks. *2019 Communication and Information Technologies Conference Proceedings, KIT 2019 - 10th International Scientific Conference*. DOI: 10.23919/KIT.2019.8883481.
- Espinosa, J. and Vandewalle, J. (2000). Constructing fuzzy models with linguistic integrity from numerical data-AFRELI algorithm. *IEEE Transactions on Fuzzy Systems*, 8(5):591–600. DOI: 10.1109/91.873582.
- Firdous, S. N., Baig, Z., Valli, C., and Ibrahim, A. (2017). Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (Smart-*

- Data), pages 748–755. DOI: 10.1109/iThings-GreenCom-CPSCoM-SmartData.2017.115.
- Haripriya, A. P. and Kulothungan, K. (2019). 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*, 2019(1). DOI: 10.1186/s13638-019-1402-8.
- Hazra, A., Rana, P., Adhikari, M., and mgoth, T. A. (2023). Fog computing for next-generation internet of things: Fundamental, state-of-the-art and research challenges. *Computer Science Review*, 48:100549. DOI: 10.1016/j.cosrev.2023.100549.
- Hwang, K., Lee, J. M., Jung, I. H., and Lee, D.-h. H. (2019). Modification of Mosquitto Broker for Delivery of Urgent MQTT Message. *2019 IEEE Eurasia Conference on IOT, Communication and Engineering, ECICE 2019*, pages 166–167. DOI: 10.1109/ECICE47484.2019.8942800.
- Jang, J., Sun, C., and Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing—A Computational Approach to Learning and Machine Intelligence. *1482 Book Reviews IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, 42(10):1482–1484. Book.
- Javaheri, D., Gorgin, S., Lee, J.-A., and Masdari, M. (2023). Fuzzy logic-based ddos attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives. *Information Sciences*, 626:315–338. DOI: 10.1016/j.ins.2023.01.067.
- Kepceoglu, B., Murzaeva, A., and Demirci, S. (2019). Performing energy consuming attacks on IoT devices. *27th Telecommunications Forum, TELFOR 2019*, pages 19–22. DOI: 10.1109/TELFOR48224.2019.8971102.
- Kerimkhulle, S., Dildebayeva, Z., Tokhmetov, A., Amirova, A., Tussupov, J., Makhazhanova, U., Adalbek, A., Taberkhan, R., Zakirova, A., and Salykbayeva, A. (2023). Fuzzy logic and its application in the assessment of information security risk of industrial internet of things. *Symmetry*, 15(10). DOI: 10.3390/sym15101958.
- Khwanrit, R., Kittipiyakul, S., Kudtonagngam, J., and Fujita, H. (2018). 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*, pages 3–8. DOI: 10.1109/ICESIT-ICICTES.2018.8442066.
- Lakshminarayana, S., Praseed, A., and Thilagam, P. S. (2024). Securing the IoT Application Layer from an MQTT Protocol Perspective: Challenges and Research Prospects. *IEEE Communications Surveys Tutorials*, pages 1–1. DOI: 10.1109/COMST.2024.3372630.
- Lee, C. C. (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part I. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):404–418. DOI: 10.1109/21.52551.
- Lekić, M., Galić, J., and Matić, S. (2019). An IoT Solution for Secured and Remote Sound Level Monitoring. *2019 18th International Symposium INFOTEH-JAHORINA, INFOTEH 2019 - Proceedings*, 1-(March):20–22. DOI: 10.1109/INFOTEH.2019.8717759.
- Li, F., Shi, Y., Shinde, A., Ye, J., and Song, W. (2019). Enhanced cyber-physical security in internet of things through energy auditing. *IEEE Internet of Things Journal*, 6(3):5224–5231. DOI: 10.1109/JIOT.2019.2899492.
- Liao, Z., Lu, X., Yang, T., and Wang, H. (2009). Missing data imputation: A fuzzy k-means clustering algorithm over sliding window. *6th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2009*, 3:133–137. DOI: 10.1109/FSKD.2009.407.
- Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., and Zhao, W. (2017). 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)*, 4(5):1125–1142. DOI: 10.1109/JIOT.2017.2683200.
- Mamdani, E. (1974). Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121:1585–1588(3). Available at <https://digital-library.theiet.org/content/journals/10.1049/piee.1974.0328>.
- Mamdani, E. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13. DOI: 10.1016/S0020-7373(75)80002-2.
- Mathur, S., Verma, A., and Srivastav, G. (2023). Analysis the three layers of computing: Cloud, fog edge. In *2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, pages 463–466. DOI: 10.1109/CICTN57981.2023.10140951.
- Naik Nitin (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*. DOI: 10.1109/SysEng.2017.8088251.
- Nazir, S. and Kaleem, M. (2019). Security with MQTT. *2019 International Conference on Information Science and Communication Technology (ICISCT)*, pages 1–5. DOI: 10.1109/CISCT.2019.8777403.
- Osanaie, O., Chen, S., Yan, Z., Lu, R., Choo, K. K. R., and Dlodlo, M. (2017). From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework. *IEEE Access*, 5:8284–8300. DOI: 10.1109/ACCESS.2017.2692960.
- Palmieri, A., Prem, P., Ranise, S., Morelli, U., and Ahmad, T. (2019). MQTTSA: A Tool for Automatically Assisting the Secure Deployments of MQTT Brokers. *2019 IEEE World Congress on Services (SERVICES)*, 2642-939X:47–53. DOI: 10.1109/services.2019.00023.
- Pappis, C. P. and Siettos, C. I. (2005). Fuzzy reasoning. In *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pages 437–474. Search Methodologies. Springer, Boston, MA. DOI: 10.1007/0-387-28356-0_15.
- Pedrycz, W. and Gomide, F. (1998). An Introduction to Fuzzy Sets: Analysis and Design. *MIT Press, Cambridge*. DOI: 10.7551/mitpress/3926.001.0001.
- Peralta, G., Iglesias-Urkia, M., Barcelo, M., Gomez, R., Moran, A., and Bilbao, J. (2017). 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, pages 1–6. DOI: 10.1109/ECMSM.2017.7945879.
- Potrino, G., De Rango, F., and Santamaria, A. F. (2019). Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker. *IEEE Wireless Communications and Networking Conference, WCNC*, 2019-April:1–6. DOI: 10.1109/WCNC.2019.8885553.
- Roohi, A., Adeel, M., and Shah, M. A. (2019). DDoS in IoT: A roadmap towards security countermeasures. *ICAC 2019 - 2019 25th IEEE International Conference on Automation and Computing*, 1(1):1–6. DOI: 10.23919/ICAC.2019.8895034.
- Sangodoyin, A., Modu, B., Awan, I., and Pagna Disso, J. (2018). 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*, pages 436–443. DOI: 10.1109/FiCloud.2018.00069.
- Selvachandran, G., Quek, S. G., Lan, L. T. H., Son, L. H., Long Giang, N., Ding, W., Abdel-Basset, M., and Albuquerque, V. H. C. (2019). A New Design of Mamdani Complex Fuzzy Inference System for Multi-attribute Decision Making Problems. *IEEE Transactions on Fuzzy Systems*, 6706(c):1–1. DOI: 10.1109/tfuzz.2019.2961350.
- Shah, B., Iqbal, F., Abbas, A., and Kim, K. I. (2015). Fuzzy logic-based guaranteed lifetime protocol for real-time wireless sensor networks. *Sensors (Switzerland)*, 15(8):20373–20391. DOI: 10.3390/s150820373.
- Shi, Y., Li, F., Song, W. Z., Li, X. Y., and Ye, J. (2019). Energy audition based cyber-physical attack detection system in IoT. *ACM International Conference Proceeding Series*. DOI: 10.1145/3321408.3321588.
- Shukla, P., Krishna, C. R., and Patil, N. V. (2024). IoT traffic-based DDoS attacks detection mechanisms: A comprehensive review. *J Supercomput*, 80:9986–10043. DOI: 10.1007/s11227-023-05843-7.
- Sikora, M., Gerlich, T., and Malina, L. (2019). On Detection and Mitigation of Slow Rate Denial of Service Attacks. *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, 2019-Octob:0–4. DOI: 10.1109/ICUMT48472.2019.8970844.
- Suguna, M., Ramalakshmi, M. G., Cynthia, J., and Prakash, D. (2018). A survey on cloud and internet of things based healthcare diagnosis. *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018*, 1-:1–4. DOI: 10.1109/CCAA.2018.8777606.
- Surya Kartika, L. G., Rinatha, K., Atmojo, Y. P., and Wayan Sukadana, I. (2019). Green Monitoring System for Energy Saving in Accommodation Services. *2019 1st International Conference on Cybernetics and Intelligent System, ICORIS 2019*, 1(August):73–78. DOI: 10.1109/ICORIS.2019.8874919.
- Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:116–132. DOI: 10.1109/TSMC.1985.6313399.
- Tian, G. Y. (2020). An Intrusion Detection System Against DDoS Attacks in IoT Networks. *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1–6. DOI: 10.1109/CCWC47524.2020.9031206.
- Toldinas, J., Lozinskis, B., Baranauskas, E., and Dobrovolskis, A. (2019). MQTT Quality of Service versus Energy Consumption. *2019 23rd International Conference Electronics*, pages 1–4. DOI: 10.1109/ELECTRONICS.2019.8765692.
- Vailshery, L. S. (2024). Number of IoT connections worldwide 2022-2033. Available at: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/> Accessed in September 2024.
- Vishwakarma, R. and Jain, A. K. (2020). A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommunication Systems*, 73(1):3–25. DOI: 10.1007/s11235-019-00599-z.
- Xiaoyan, Z. (1996). ZY. *Institute of Juzzy Systems and Knowledge Engineering Republic of China*, 77:175–190. Book.
- Xu, C., Xiong, Z., Zhao, G., and Yu, S. (2019). An Energy-Efficient Region Source Routing Protocol for Lifetime Maximization in WSN. *IEEE Access*, 7:135277–135289. DOI: 10.1109/access.2019.2942321.
- Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., and Al-Hatmi, R. (2018). Internet of Things: Survey and open issues of MQTT protocol. *Proceedings - 2017 International Conference on Engineering and MIS, ICEMIS 2017*, 2018-Janua:1–6. DOI: 10.1109/ICEMIS.2017.8273112.
- Yousaf, R., Ahmad, R., Ahmed, W., and Haseeb, A. (2017). Fuzzy Power Allocation for Opportunistic Relay in Energy Harvesting Wireless Sensor Networks. *IEEE Access*, 5:17165–17176. DOI: 10.1109/ACCESS.2017.2743063.
- Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-I. *Information Sciences*, 8(3):199–249. DOI: 10.1016/0020-0255(75)90036-5.
- Zimmermann, H. J. (2001). *Fuzzy Set Theory, and its applications*. Springer Netherlands, Dordrecht, 4 edition edition. Book.